TRANSPARENT DMA DATA ACQUISITION SYSTEM FOR REAL TIME APPLICATION

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

By

SQN. LDR. RAJIV SHARMA

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1985

12/4/85

CERTIFICATE

It is certified that this work entitled 'TRANSPARENT DMA DATA ACQUISITION SYSTEM FOR REAL TIME APPLICATION' has been carried out under my supervision by SQN LDR RAJIV SHARMA and has not been submitted elsewhere for a degree.

R. Raghuram

(R. Raghuram)
Assistant Professor
Department of Electrical Engineering
Indian Institute of Technology
Kanpur

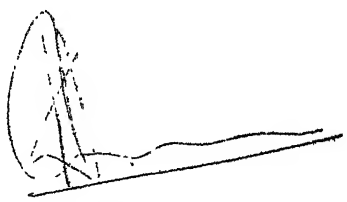
27/4/85

EE-1985-M-SHA-TRA

ACKNOWLEDGEMENT

I wish to express my sincere thanks to my thesis supervisor Dr. R. Raghuram for his guidance and encouragement at various stages of this thesis. The guidance given by him and the useful discussions I had with him has helped immensely in the completion of this thesis.

I am thankful to Mr. Prem Malhotra for his suggestions and the useful discussions I had with him on certain hardware aspects.



RAJIV SHARMA

CONTENTS

ABSTRACT

In any microcomputer system, used in Real time processing, DATA acquisition could be put under three main headings

a) Program Controlled
b) Interrupt - Program Controlled
c) DMA or Hardware Controlled

In the first two, considerable CPU time is wasted in DATA transfer. DMA allows to handle data transaction with external devices, without actually involving the CPU. Thus achieving high Data transfer rate.

There are three common ways of achieving DMA transfers. These are burst DMA, cycle steal DMA and transparent DMA. During Burst DMA operation CPU cannot execute its program. Cycle stealing DMA does allow the processor to work concurrently with the DMA operation, but not during, when actual DATA is being transferred. Transparent DMA removes this restriction also.

In this project, for Intel 8085A microprocessor a special purpose hardware has been designed for transparent DMA. In the op-code fetch (OF) machine cycle of the 8085A, after the third state data bus floats and the address bus contains unspecified Data. Transparent DMA has been done during these states. This is referred to as transparent DMA since it is completely transparent to the processor. It does not interfere with or slow down the processor's own program execution.

Transparent DMA is best suited where processing time is critical like in Signal Processing.

CHAPTER 1

INTRODUCTION

1.1 THE CONCEPT OF DATA ACQUISITION:

Data acquisition plays a vital role in microcomputer systems, especially those used in real time processing. This is because during data acquisition the CPU's normal processing activity gets interrupted. In certain applications e.g. signal processing where the processing time is very critical this interruption should be minimised. The duration of the interruption depends on how the data acquisition is being done. There are many ways in which data acquisition from an external device to the microcomputer system can be accomplished; but they all fall into the following three categories:

i) Program Controlled

ii) Interrupt Program Controlled

iii) D.M.A. or Hard ware Controlled.

With Program controlled data acquisition, it is completely under the control of the microprocessor. In this case the data acquisition operation is carried out only when the microprocessor encounters a data IN instruction,

in whatever program it is executing. One can have two kinds of program controlled data acquisition - Unconditional or Conditional. In unconditional program controlled data acquisition, the microprocessor receives data from an IN-port without making any attempt to ascertain wnether or not the port is ready with the correct data. In conditional program control, data acquisition from an IN-port is conditioned on the port being ready for it. The IN-port signifies its readiness by setting appropriate flag bits in the status register where this status register is itself read as another IN-port.

Therefore, the problems associated with Program Controlled data acquisition should be quite evident. The Handshaking, i.e. testing status bits, would require software and the time, and the hardware required for this would be considered as data acquisition overhead. A more serious problem is that of timing. The microprocessor must service the port at least the same speed at which the external device uses this port. The status bits should be tested in a program sequence tight enough so that any 'request for service' from an IN-port is serviced before the external device connected to that port generates the next request.

In time critical applications this can pose quite a few problems in writing and executing the program being executed by the C.P.U. Moreover the overhead involved increases with the number of IN-ports in the system even when most of these devices require very intermittent service.

A more efficient method of data acquisition is Interrupt-Program Controlled. In this case an IN-port requiring service ('Data Transation' with system) signals its need to the microprocessor by driving an interrupt input of the microprocessor high. A microprocessor which is interrupted, suspends its normal program execution. Control is then transferred to an interrupt service routine. In this case, this routine is expected to service the IN-port, after which control is returned to the interrupted routine. In this case, external hardware signals the microprocessor directly when one needs to service an IN-port but the actual data transfer is still handled by the microprocessor using the interrupt service routine. Unlike program controlled data acquisition the microprocessor does not have to do the initiation itself. The amount of work to be done by the microprocessor and the data acquisition overhead is correspondingly less. Interrupt controlled data acquisition allows the system to initiate data acquisition using hardware

external to the microprocessor but the actual data transfer still involves the C.P.U. (i.e. data will be input from port to some register in the C.P.U. then register to memory).

1.2 DMA DATA ACQUISITION:

Let us consider an application where the microprocessor system is interfaced to a peripheral device, e.g. a computer. At certain instants the computer dumps N data bytes into consecutive RAM locations of the microprocessor system. Assume that the computer also signals the start of the data transfer in some fashion possibly by using an interrupt line.

If we were using interrupt controlled data acquisition we will do the following:

i) When we get the 'intialization' interrupt, we initialize a data counter with the starting address of the RAM table where the data is to be stored.

ii) When the computer has a data byte ready to send, it presents an interrupt request to the microprocessor system. The microprocessor acknowledges the interrupt and executes an interrupt service routine which loads the data byte in the location pointed to by the data counter. The data counter is then incremented. (If the number of data bytes transferred

is less than N, the microprocessor would expect more interrupts of this kind. Otherwise, it would expect an 'initialization' interrupt.

Examining (ii) closely we find that the only difference between successive executions of the interrupt service routine is that the data counter contents are different. All the other instructions of the service routine will essentially do the same thing every time the routine is executed. Moreover every time an interrupt request is received, some time is spent in acknowledging it, calling the proper interrupt service routine, and returning from that routine. All these overheads will not be there if we can do the following:

(i) Initialize the RAM storage table when we get the initialization interrupt. The service routine for that interrupt should find out (may not always be necessary) what is the value of N- the number of data bytes to be transferred.

(ii) Somehow handle the N data transfers without directly invoking the microprocessor (no interrupts or program controlled checks for data).

This is exactly what is done in Direct Memory Access data acquisition.Unlike Interrupt data acquisition, the actual

data transfers occur directly between the IN-port and memory. (Interrupt controlled data acquisition will route the data through the processor using instructions like IN port/MOV r,M etc.) This implies that if address and data bus are avoided the transfer of data can even work simultaneously with the processor executing an instruction of its program.

There are three common ways of achieving D.M.A. transfers. These are

i) Burst D.M.A.

ii) Cycle Steal D.M.A.

iii) Transparent D.M.A.

Actual data transfer is carried out under the supervision of the D.M.A. controller. D.M.A. controller contains mainly four registers, which are required to be initialised by the microprocessor before the actual D.M.A. transfer.

i) An Address Register - which contains the address of the next memory location to be accessed.

ii) A Counter Register - has the information how many bytes are to be transferred.

iii) A Control Register - which identifies the direction of data flow, and is used to start, stop or otherwise control D.M.A. operation.

iv) A Status Register - which is used to identify the status of any D.M.A. operation that may be in progress.

A Burst DMA operation needs to execute following steps -

i) Whenever an external device wants to do data transation with the microprocessor it puts the request to DMA controller, which in turn raises the Hold request to microprocessor.

ii) The microprocessor responds to the Hold request by returning a Hold Acknowledge (HLDA) to the DMA controller when it can indeed acknowledge HRQ. (This typically happens after the execution of the current machine cycle being executed). It also simultaneously floats its address and data buses and the appropriate control lines, e.g. $\overline{RD}$, $\overline{WR}$ and IO/$\overline{M}$.

iii) Now DMA controller becomes the bus master of buses and provides address on the address bus. Depending on the Read or Write operation to be performed appropriate control signals are provided by the DMA controller for data transfer.

iv) After transfer of one byte the DMA controller increments its address register to point to the next memory location and decrements its counter register. If the terminal count has been reached - i.e. all the required

number of bytes have been transferred - the DMA controller interrupts the microprocessor to indicate that the DMA operation has been completed.

In case of Burst DMA operation microprocessor is 'held-off' and the data transfer goes on continuously till the time external device transferring the data keeps the DMA request high, while a Burst DMA operation is going on the microprocessor is actually 'held-off' and is sitting idle because it can not execute its program during this time. If the Burst DMA operation is a long one this by itself can pose problems.

The major difference between Interrupt controlled data acquisition and Burst DMA data acquisition is that unlike the former, the latter does not need the active participation of the microprocessor. In Interrupt controlled data acquisition, the data bytes being transferred actually pass through the internal registers of the CPU. In Burst DMA this does not happen; in this case the processor merely initiates the operation and then stands by and waits till the DMA operation is over.

Cycle stealing DMA is very similar to Burst DMA with one major difference. Unlike Burst DMA cycle stealing DMA occurs concurrently with other processing being carried

out by the microprocessor. The execution steps are similar to those given earlier for Burst DMA except that if terminal count has not been reached in step (iv), steps (i) to (iv) are repeated. This in effect, means that the DMA controller removes its HOLD Request (HRQ) periodically after each data byte has been transferred and repeates the request when the next data byte to be transferred is available. This gives the processor time to get back the control of the various buses so that it can execute its own machine cycles, at least until the next HOLD Request (HRQ) arrives.

Cycle stealing DMA will be slower than Burst DMA because the processor gets back its control in between byte transfer. It however has the advantage that it allows the processor to continue its own program execution, albeit at a slower rate.

Even though cycle stealing DMA does allow the processor to work concurrently with the DMA operation it still interferes somewhat with the processor's operation, i.e. when an actual byte transfer is taking place. Transparent DMA removes this restriction also, even though it is the slowest D.M.A. mode.

If one examines the machine cycles of microprocessors' one finds that there are clock states during some machine

cycles when the processor does not use the data and address bus. During these states, the processor either floats these buses or they contain unspecified. (In the latter case, assume that sufficient external logic is provided to float these buses during these states). Using these states (with buses floated by external logic, if necessary), a DMA operation can be done. This is referred to as Transparent DMA since it is completely transparent to the processor. It does not interfere with or slow down the processor's own program execution. Transparent DMA requires extra logic to detect the states when only internal processing is being carried out by the microprocessor.

Burst DMA is the fastest, whereas Transparent DMA is the slowest, conversely, Burst DMA interferes most with the processors own operation whereas Transparent DMA creates the least interference.

1.3 PROJECT BRIEF:

INTEL 8085A microprocessor's Op-Code Fetch (OF) machine cycle has either 4 clock-states or 6 clock states. Out of total 246 announced instructions, only 23 instructions are that of 6 clock-state type, OF. In the OF machine cycle during T4 (T_5 and T6 also, if they exist) data bus floats and the address bus contains unspecified data.

In this project, for INTEL 8085A microprocessor a special purpose hardware has been designed for Transparent DMA during OF machine cycle. Data acquisition has been done from the external logic by putting the microprocessor in Hold condition for two clock states starting from T4 state of OF machine cycle. Hardware has been designed to work with SDK-85 Kit.

In the applications like signal processing, where the processing time is very critical, and no time can be wasted for data acquisition, Transparent DMA data acquisition system is best suited. To examine the performance of the designed system, a second order low pass Butterworth digital filter has been implemented software wise. During the filter program execution simultaneously input data acquisition could be done without interrupting the processor.

Intel 8085A microprocessor has a clock frequency of 3.125 MHz (clock period 330 nsec.) and the same frequency has been used in the hardware designed for Transparent DMA. Intel 8257 is a programmable DMA controller device, which needs 4 clock states with a minimum period of 320 nsec, to transfer one data byte whereas the hardware designed for Transparent DMA needs only 2 clock states out of these two

clock states one state is utilised from OF machine cycle (T4), thus it needs only one clock state to transfer one data byte.

In 8257, Address register, Terminal count register, and control register is required to be initialised before data transfer. The initialisation is required for every block of data to be transferred. Every initialisation takes about 30 µsec. However the hardware for Transparent DMA does not require any initialisation.

For every 256 bytes of data transfer from external logic to the memory Intel 8257 would take 284 µsec. extra, compared to the designed Transparent DMA system.

1.4 REAL TIME OPERATION:

Real time operation can be defined as - 'Solving problems in real time. More precisely, processing data in time with a physical process so that the results of the data processing are useful in guiding the physical operation.

CHAPTER 2

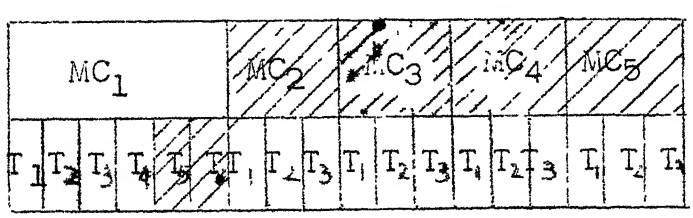
HARDWARE TIMING

2.1 INTRODUCTION:

In this chapter first the timings of OF machine cycle, memory write cycle and Hold state of 8085A microprocessor have been discussed. Next, timings for the Intel 2114AL-1 have been given which is used to store input data. Then various timings adjusted in the external hardware to perform transparent D.M.A. during OF machine cycle have been discussed.

2.2 8085A TIMING:

An 8085A instruction's execution is timed by a sequence of machine cycles, each of which is divided into clock periods. An instruction is executed in from one to five machine cycles, labeled MC_1, MC_2, MC_3, MC_4 and MC_5. The first machine cycle of any instruction's execution has either four or six clock periods. Subsequent machine cycles have three clock periods only. This may be illustrated as follows.



where MC is shaded, the entire machine cycle is optional. Where T is sheded, the clock perioa is optional within its machine cycle.

ALE signal identifies the start of a new machine cycle. There are seven different type of machine cycles in 8085A. They can be differentiated by the state of the three status lines ($IO/\overline{M}$, S_0, and S_1) and the three control signals ($\overline{RD}$, $\overline{WR}$, and $\overline{INTA}$), as shown in Table 2.1.

Table 2.1: 8085A Machine Cycle Chart.

Machine cycle		STATUS			CONTROL		
		$IO/\overline{M}$	S_1	S_0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
Op-Code Fetch	(OF)	0	1	1	0	1	1
Memory Read	(MR)	0	1	0	0	1	1
Memory Write	(MW)	0	0	1	1	0	1
I/O Read	(IOR)	1	1	0	0	1	1
I/O Write	(IOW)	1	0	1	1	0	1
INTR Acknowledge	(INA)	1	1	1	1	1	0
BUS Idle	(BI)DAD	0	1	0	1	1	1
	INA	1	1	1	1	1	1
	HALT	T_s	0	0	T_s	T_s	1

O=logic '0', 1=logic '1', T_s= High impedance,

2.2.1 Op-Code Fetch:

The most important aspect of the OF machine cycle is the fact that it has either four or six clock periods, as against three for all subsequent machine cycles. The OF machine cycle must have atleast four clock periods, since the fourth clock period is needed to decode the instruction object code which has been fetched. During T_4, CPU also decides whether to enter T_5 on the next clock or to start on new machine cycle and enter T_1.

At the end of the first clock period, AD_0-AD_7 is floated transiently; then it is turned around to act as a Data Input Bus. $\overline{RD}$ is pulsed low to strobe data onto the Data Bus. The memory read must occur within three clock periods.

During the fourth clock period of the OF machine cycle the instruction object code is interpreted by logic of the 8085A C.P.U. Fifth and sixth clock periods are required by some instructions to execute required internal operations.

During the fourth and subsequent clock periods, AD_0-AD_7 is floated and A_8-A_{15} contains unspecified data. In Fig. 2.1 S_0 and S_1 are both high, identifying this as an op-code fetch machine cycle. IO/$\overline{M}$ is low since instruction object code is to be fetched from memory.

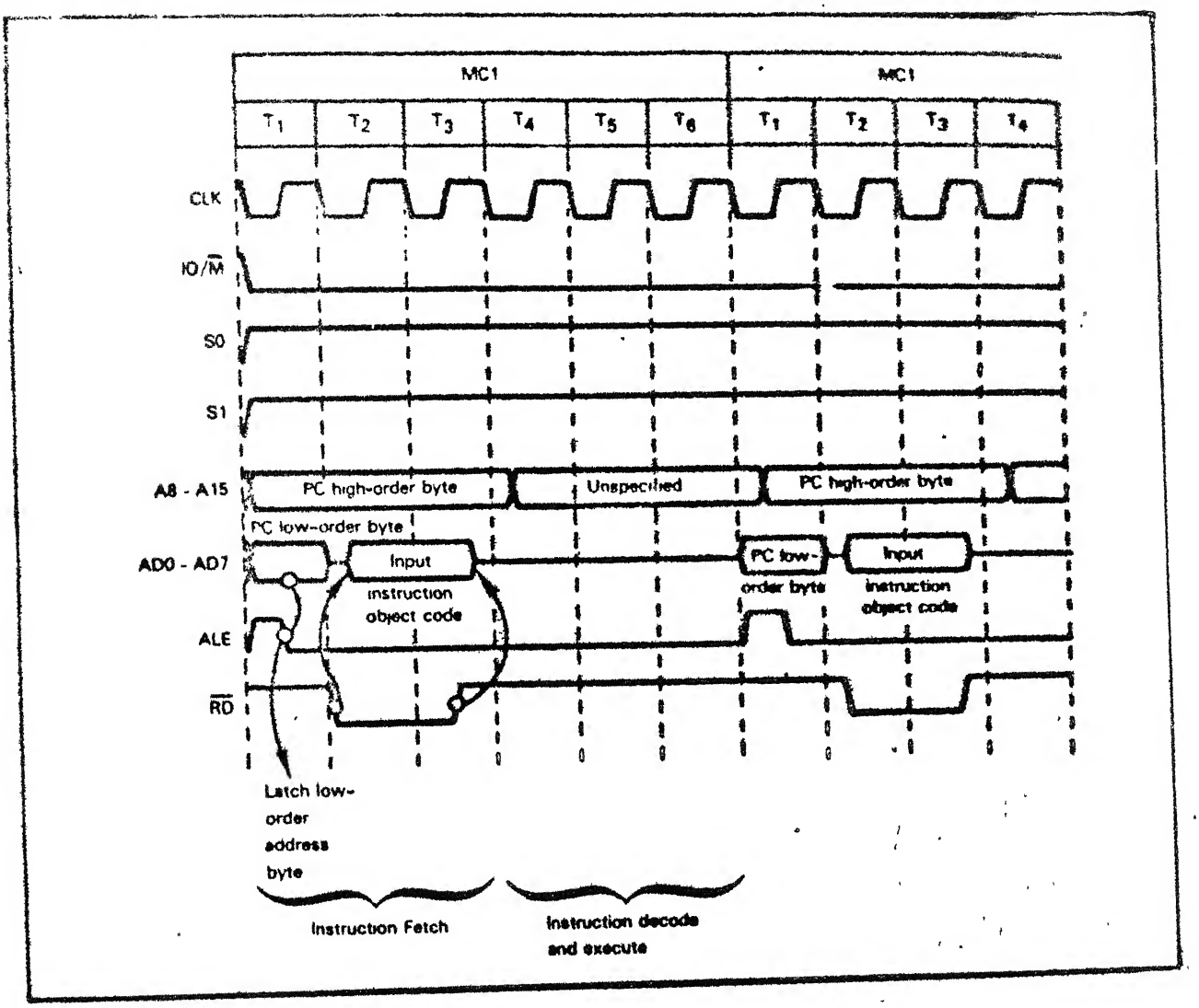


Fig. 4.1: A Six Clock Period Op-Code Fetch Machine Cycle

2.2.2 Write Cycle Timing:

Fig. 2.2 shows the timing for Memory Write machine cycle. The 8085A sends out the status during T_1 in a similar fashion to the OF cycle, except that $IO/\overline{M} = 0$, $S_1 = 0$ and $S_1 = 1$, identifying the current machine cycle as being a WRITE operation to a memory location. The address

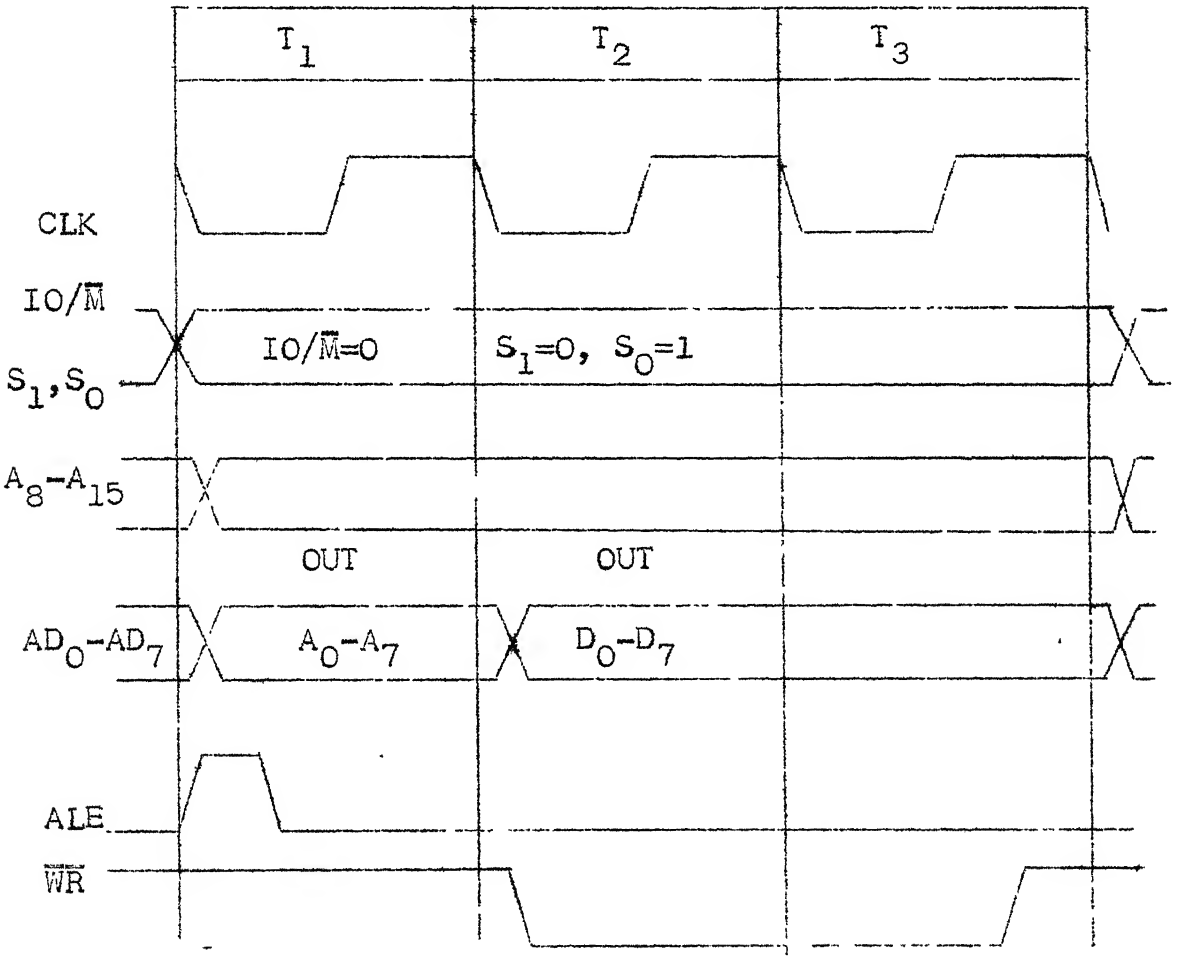


Fig. 2.2: Memory Write Machine Cycle.

is sent out during T_1 in an identical manner to OF. The Data to be written into the addressed memory location is placed on AD_0 - AD_7 at the start of T_2. The $\overline{WR}$ signal is also lowered at this time to enable the writing of the addressed memory device. During T_3, the $\overline{WR}$ line is raised, disabling the addressed memory device and thereby terminating the Write operation. The contents of address and data lines are not changed until the next T_1 which follows-directly.

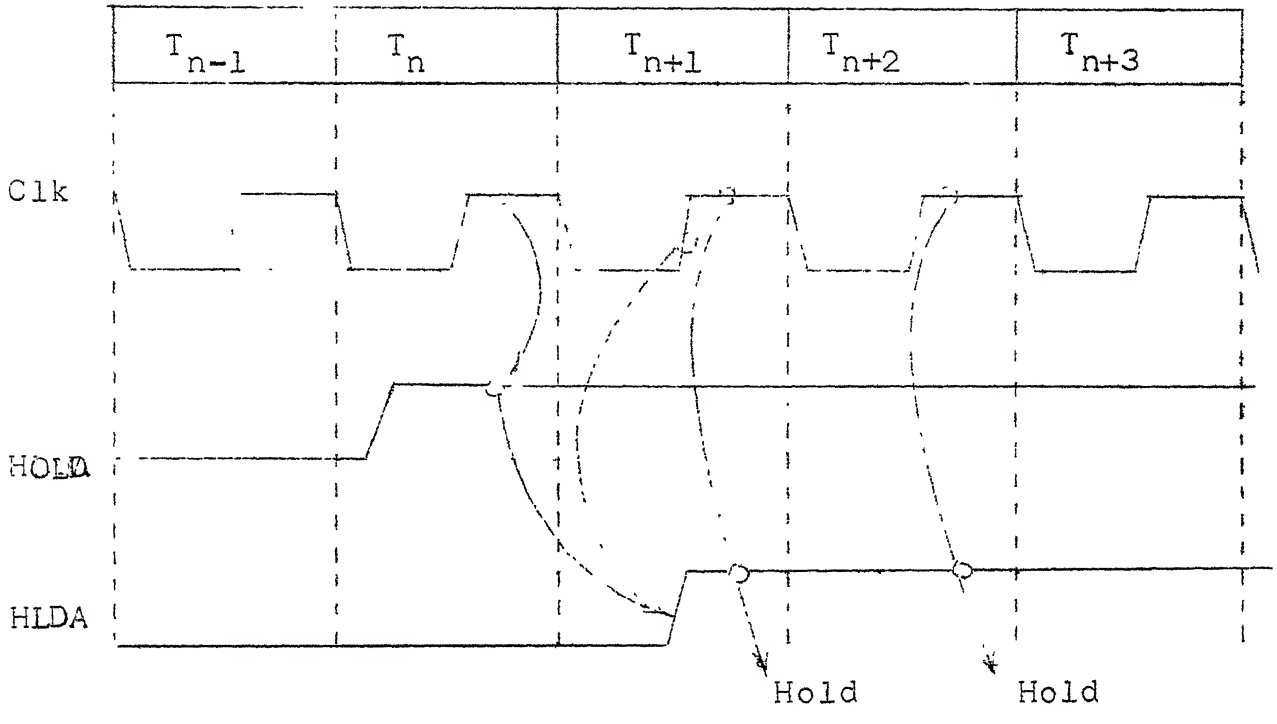
The data on AD_0-AD_7 is not guaranteed to be stable before the falling edge of $\overline{WR}$. The AD_0-AD_7 lines are guaranteed to be stable both before and after the rising edge of $\overline{WR}$.

2.2.3 The HOLD State:

8085A uses the Hold state as a means of transiently floating the system bus. During a Hold, external logic gains bus control, usually to perform direct memory operations.

External logic requests a Hold state by putting HOLD high. The microprocessor responds by entering the Hold state and making HLDA high. During a Hold state the microprocessor floats all tristate signals. However, ALE is kept low during Hold.

The 8085A has a fixed, two machine cycle sequence for Hold state initiation; it may be illustrated as follows:



During every machine cycle, Hold is sampled during T_2; if Hold is high at this time, then Hold acknowledge is made high during T_3 and the Hold state begins during T_4. Timing is illustrated in Fig. 2.3. Once the Hold is found high in T_2 then in every subsequent clock periods it will be examined until HOLD does not, go low.

During a six clock period machine cycle, if Hold is low when sampled during T_2, then Hold will be sampled again during T_4. If Hold is sampled high during T_4, then

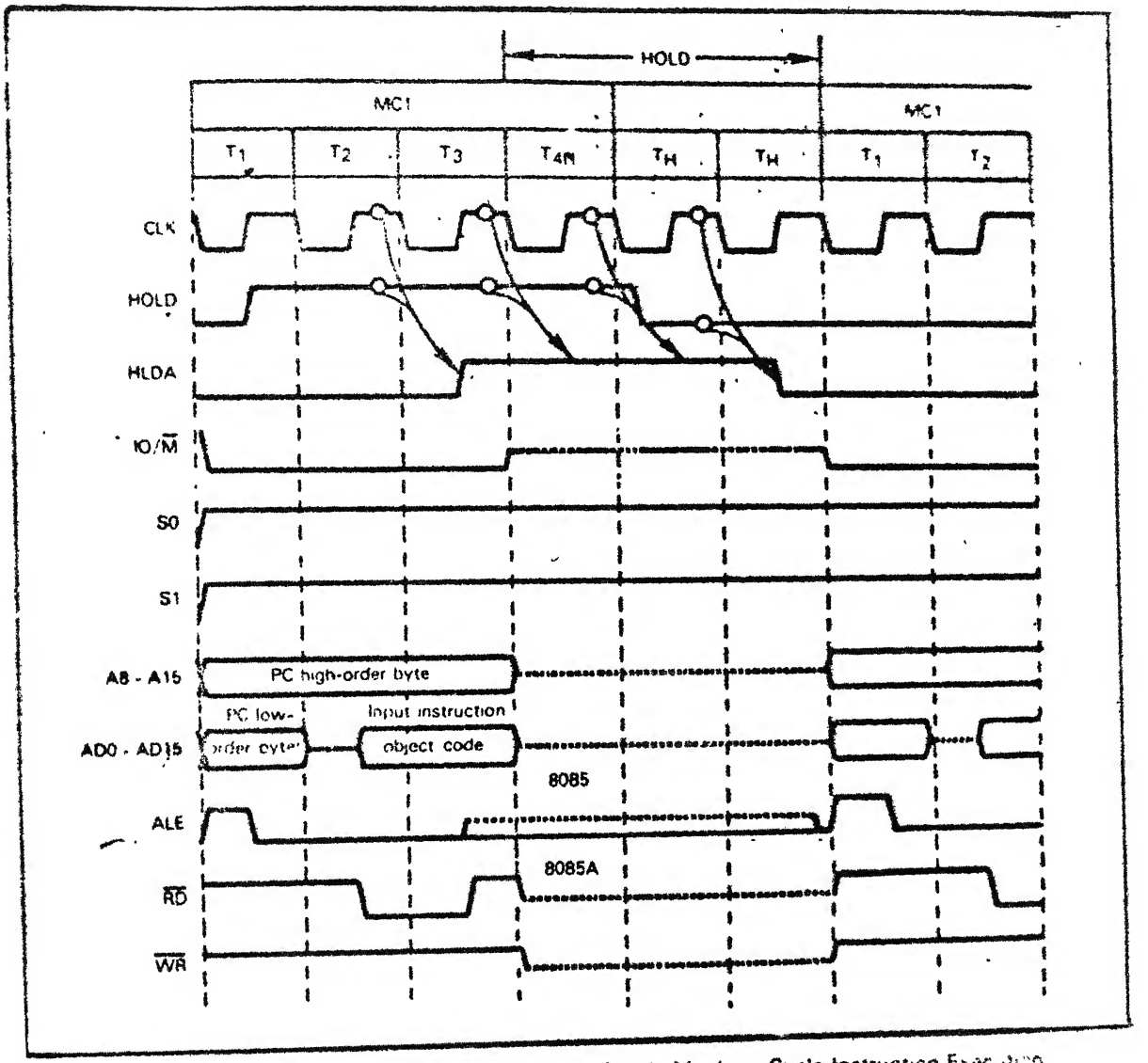
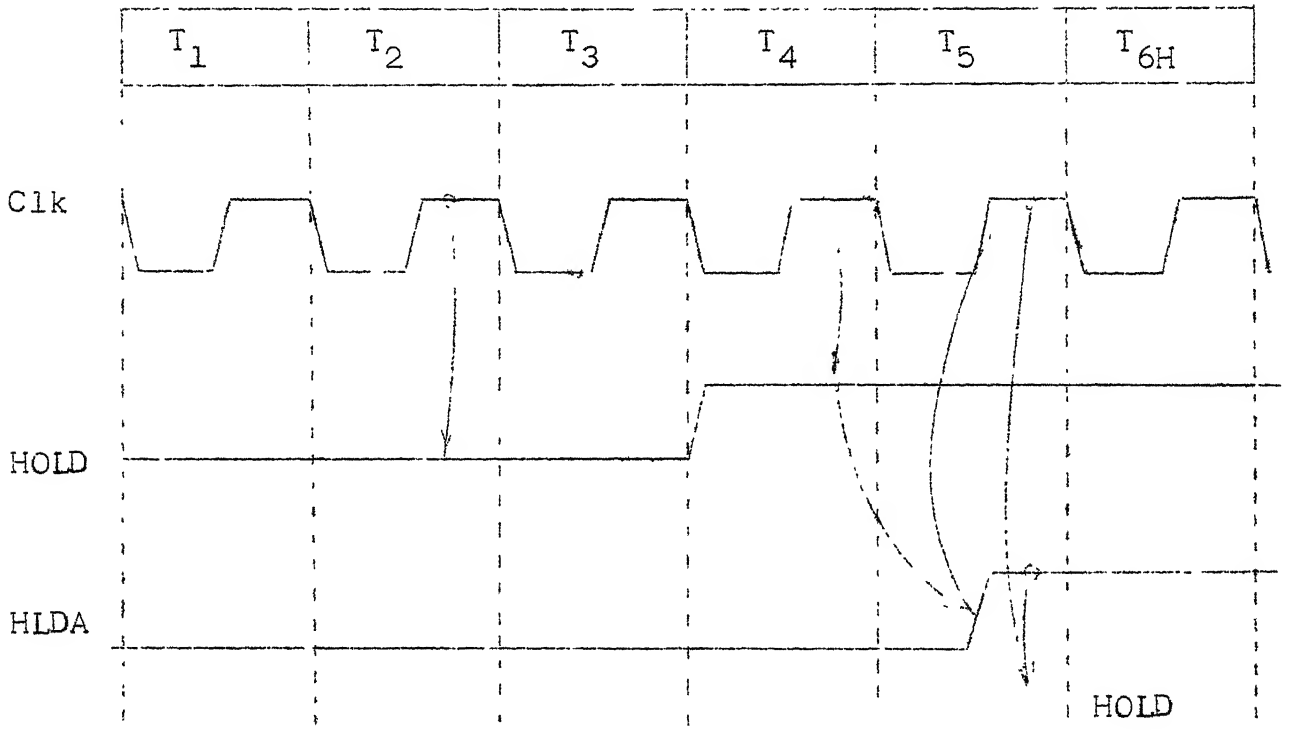


Figure 2-3 A Hold State Following a Single Machine Cycle Instruction Execution

a Hold state will initiated during T_6. This may be illustrated as follows:



A Hold state terminates two clock periods after the Hold signal goes low. There are no restrictions placed by 8085A logic on the duration of a Hold state. The Hold state lasts as long as the HOLD input is high. Fig. 2.3 illustrates a Hold state lasting three clock periods, beginning during T_4 of a four clock period machine cycle.

2.3 INTEL 2114AL-1 STATIC RAM:

The Intel 2114AL-1 is a 4096 bit static Random Access Memory organised as 1024 words by 4 bits using N-channel

silicon gate MOS technology. It requires no clocks or refreshing to operate. Data access is particularly simple since address setup times are not required. The data is readout non destructively and has the same polarity as the input data. Common Input/Output pins are provided.

LOGIC SYMBOL

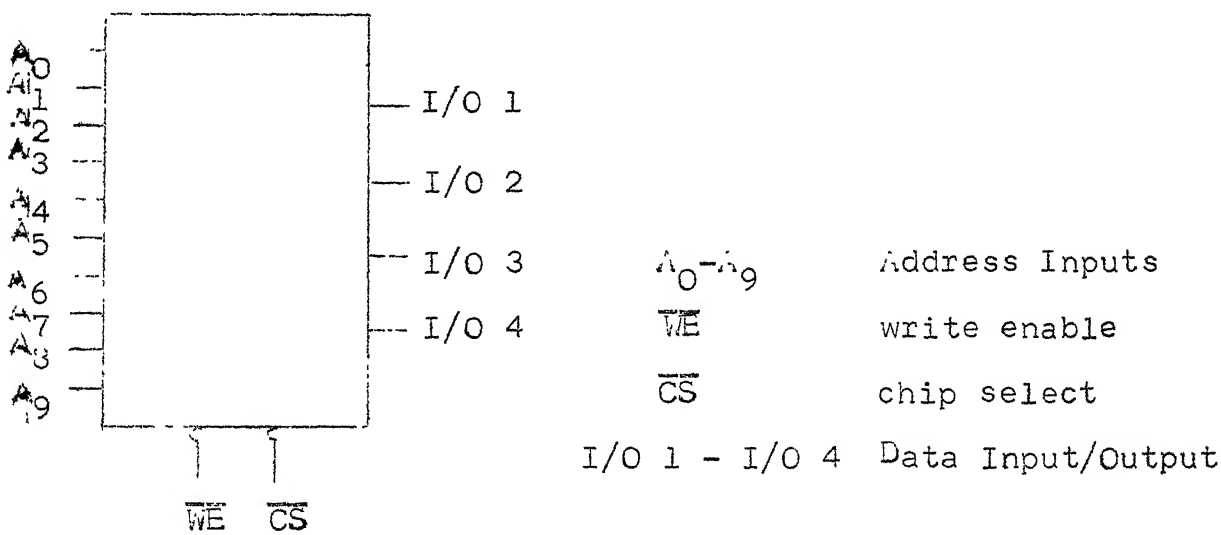
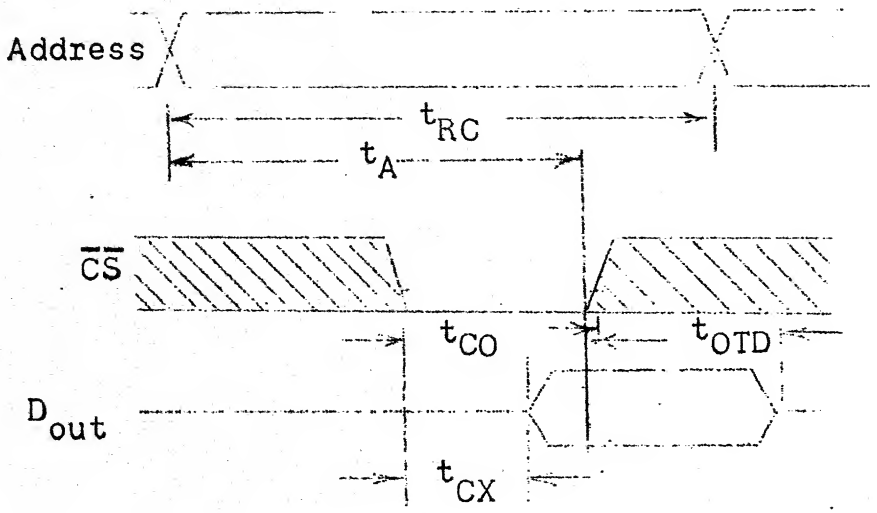


Fig. 2.4 and Fig. 2.5 give the Read and Write cycle waveforms.

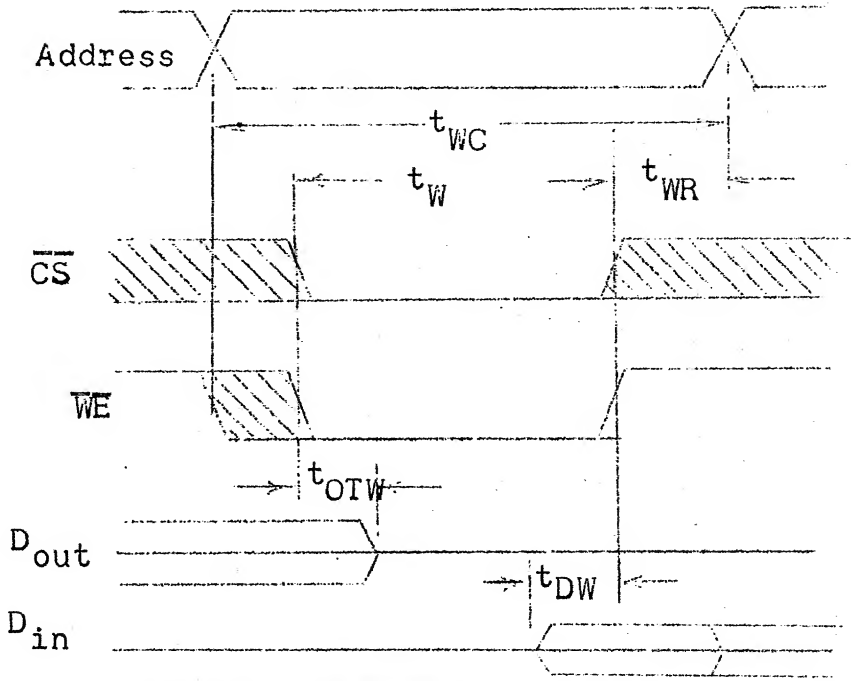
READ CYCLE



SYMBOL	PARAMETER	Min	Max
t_{RC}	Read Cycle Time	100	
t_A	Access Time		100
t_{CO}	Chip Selection to Output		70
t_{CX}	Chip Selection to Output Active	10	
t_{OTD}	Output 3-state from Deselection		30

Fig. 2.4

WRITE CYCLE



SYMBOL	PARAMETER	Min	Max
t_{WC}	WRITE CYCLE TIME	100	
t_W	Write Time	75	
t_{WR}	Write Release Time	0	
t_{OTW}	Output 3-state from Write		30
t_{DW}	Data to Write time overlap	70	

Fig. 2.5

2.4 TYPICAL TTL DELAYS:

The series 54/74 TTL family has five major divisions; Standard (SN 54/74), High Speed (SN 54H/74H), Low Power (SN 54L/74L), Schottky-Diode-Clamped (SN 54S/74S) and Low Power Schottky (SN 54LS/74LS). All the five types are compatible and interface directly with one another. Certain characteristics are common in all the five, but Propagation delay time and power dissipation/gate are different. The broad spectrum of Speed/Power combination enables to optimize all portions of a system as per required performance specifications.

Table 2.2 gives the typical propagation delay time and power dissipation for different types of TTL gate.

Table 2.2 : Propagation Delay Time in nsec.

TYPE	Propagation Delay Time		Typical Power Dissipation/Gate
	Min.	Max.	
Standard	10	19	10 mw
High Speed	6	11	22 mw
Low Power	33		1 mw
Schottky-Diode-Clamped	3	5	19 mw
Low Power Schottky	9	18	2 mw

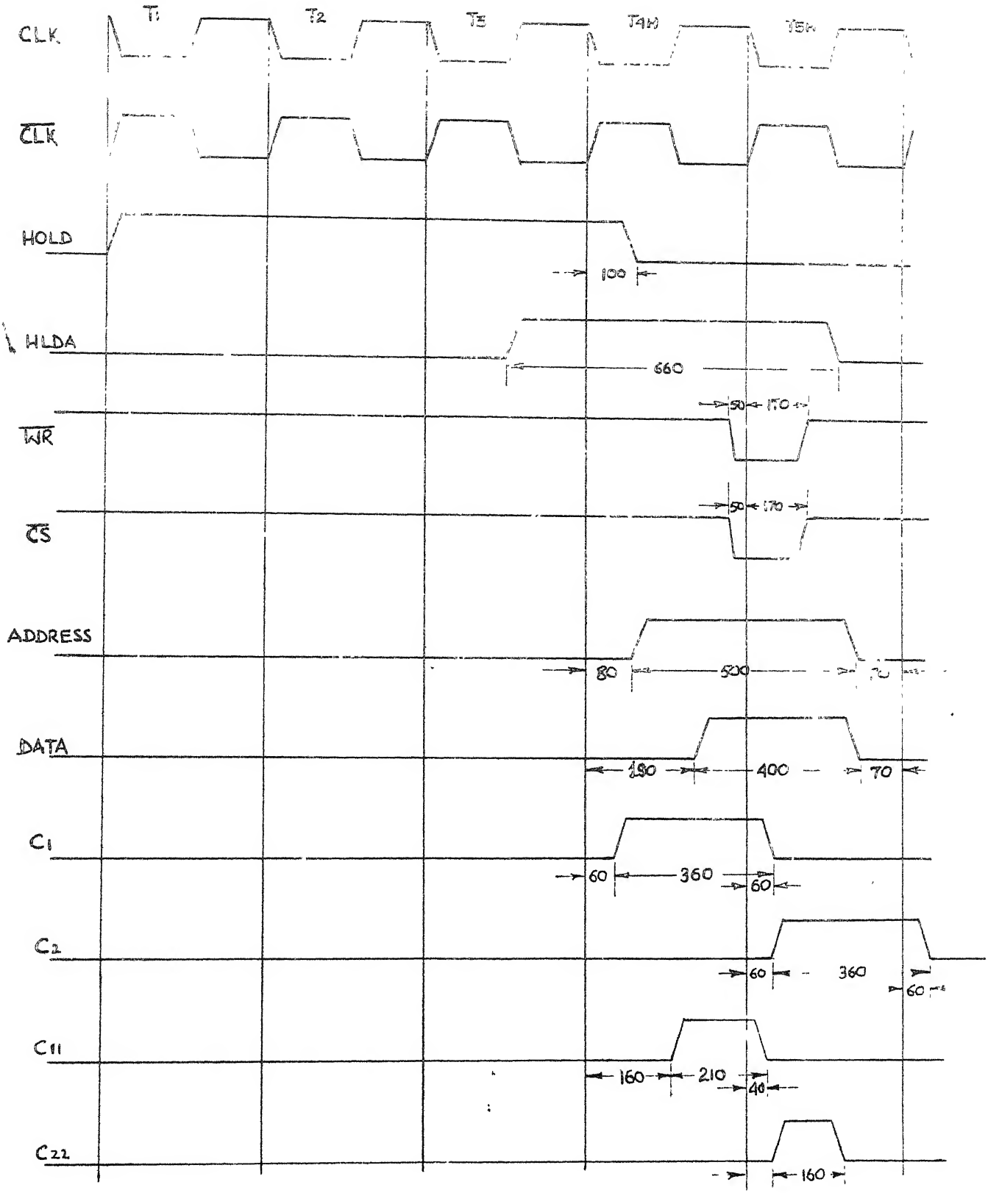
Thus it can be seen that Schottky-Diode-Clamped has the highest speed. Because of the minimal propagation delay, most of the chips used in the Hardware are of this type.

2.5 HARDWARE TIMINGS:

The transparent DMA data acquisition operation during OF machine cycle can be divided into two parts. First, putting the microprocessor in Hold condition for two states i.e. during T_4 and T_5, second, during these two states generation of appropriate address bits and control signals to store input data.

2.5.1 Hold States:

After the data conversion by ADC, in the first OF machine cycle, HOLD is raised high by the external logic. This is done right at the beginning of T_1 state. It is kept high until end of T_3 state. Microprocessor examines the Hold during T_2 and raises HLDA high from middle of T_3 state. As Hold is kept high for T_2 and T_3 states, HLDA remains high till middle of T_5 state. Thus, we get T_4 and T_5 states as Hold states. Fig. 2.6 illustrates the relative timings.



(TIME IN nSec)

FIG 2·6 HARDWARE TIMING

2.5.2 Control Signals:

A 4-bit binary counter has been used in the external logic to count the clock periods during Hold. Detailed description of the counter is given in the Chapter 3. Outputs from this counter have been hardwired in two different combinations C_1 and C_2 to identify T_{4H} and T_{5H} states.Further C_1 and C_2 have been ANDed with the system Clk and $\overline{Clk}$ respectively, and these outputs have been referred as C_{11} and C_{22}.

It is shown in the timing diagram (Fig. 2.6)that C_1 and C_2 do not go high right at the beginning of T_{4H} and T_{5H} state but after a delay of 60 nsec. This is due to the propagation delay of the various TTL gates.

Tristate buffers have been used in the external logic for data bus, address bus and control signals. These buffers are required to be enabled during the Hold state for Data transfer. Different combinations of C_1, C_2, C_{11} and C_{22} have been used to enable the buffers.

2.5.3 Address Bus Buffer:

C_1+C_{22} combination enables the address bus buffer. Buffer gets enabled after a delay of 80 nsec for 500 nsec. To avoid bus conflict between external logic and the microprocessor, it is disabled 80 nsec before the start of next machine cycle.

2.5.4 Data Bus Buffer:

$C_{11}+C_{22}$ combination is used to enable the data bus buffer. It gets enabled after 190 nsec for 400 nsec. Then, it is disabled 70 nsec before the next machine cycle starts, to avoid bus conflict.

$\overline{C_{11}}$ has been used to generate $\overline{WR}$. Two conditions are required to be met for memory write cycle. First, $\overline{WR}$ and $\overline{CS}$ should go low simultaneously. Second, $\overline{WR}$ should go low only after the data bit gets stable on the bus. Accordingly, timings for $\overline{WR}$ and $\overline{CS}$ have been adjusted. It is seen from the timing diagram(Fig. 2.6)that $\overline{WR}$ goes low after 90 nsec of Data bit goes high. $\overline{WR}$ and $\overline{CS}$ low appear only at the end of T_{4H}, for 200 nsec. During the transition of address and data bits, $\overline{WR}$ should remain high.

Timing adjustment has been the most critical part of this project. To minimise delay in the signal paths Schottky-diode-clamped type TTL ICs have been used. At certain stages signals are required to be delayed for proper adjustment in timings, e.g. C_1+C_{22} combination gives a pulse of 500 nsec duration which enables the address bus buffer. Here C_1 is delayed by 20 nsec to avoid dip in the pulse. Similarly rise time and fall time of pulses are also reduced to minimum. Hardware details about these have been given in the next chapter.

CHAPTER 3

THE HARDWARE

3.1 INTRODUCTION:

In this chapter detailed description about the hardware is presented. The functional block diagram of the system is described, which can be broken down into following four major parts.

i) Timing and Control Unit.
ii) Memory Addressing Unit.
iii) Data Conversion Unit.
iv) Data Storage Unit .

These four units have been described with the schematic diagrams. Certain special hardware features have been listed at the end.

3.2 SCHEMATIC OF HARDWARE:

Transparent DMA data acquisition system has been designed to work with the SDK-85 kit. However it can work with any system using 8085A. In the SDK-85 kit, all the control and status lines, along with demultiplexed address and data buses are available outside for system expansion.

Fig. 3.1 is a functional block diagram of the system. There are two memory blocks where the input data is stored.

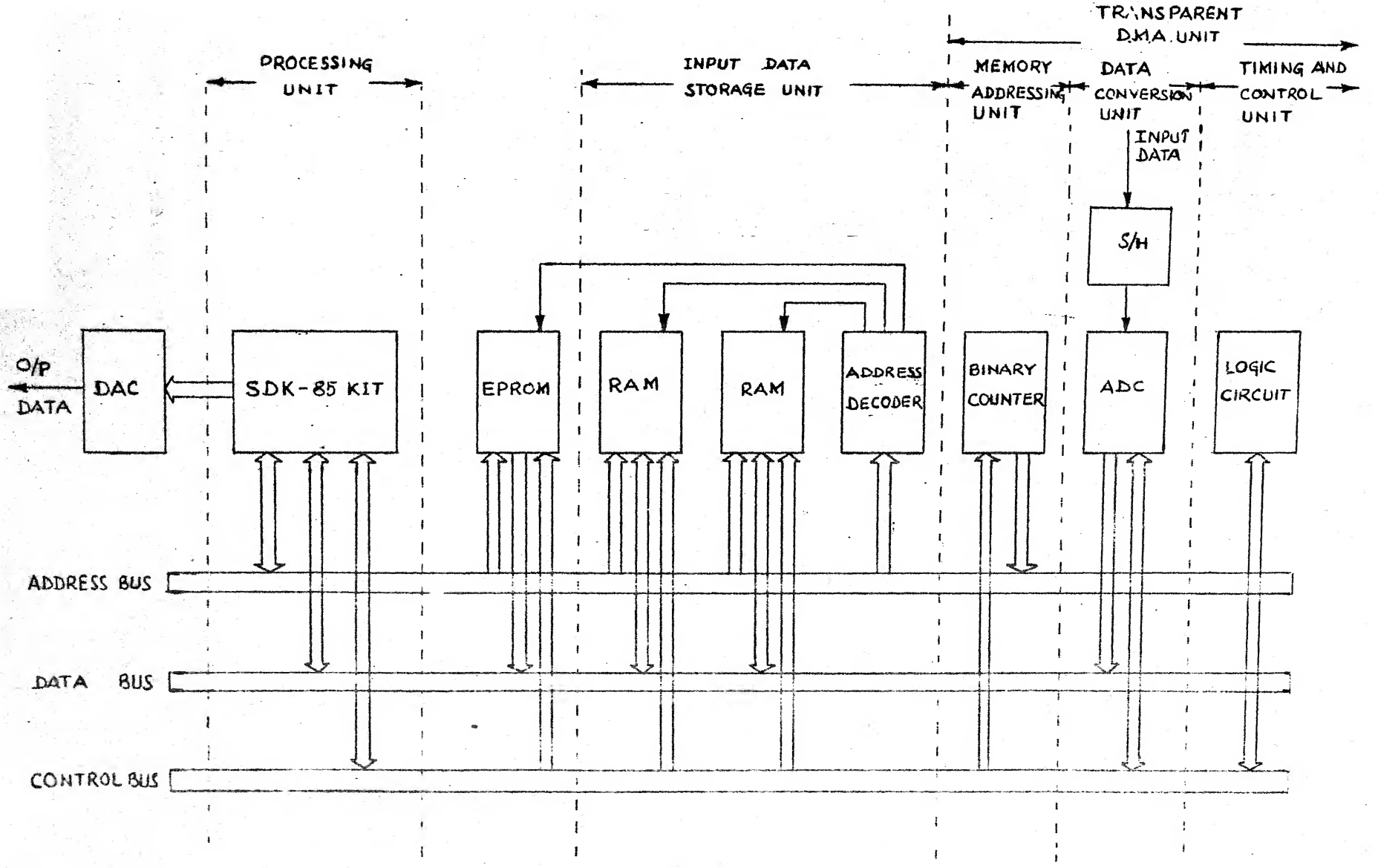


FIG 3.1 FUNCTIONAL BLOCK DIAGRAM

These memories can be accessed both by the microprocessor and the external hardware. However, at any instant of time, they can be accessed from one side only.

Input data could be either in the analog or digital form. Input if analog is given to A.D.C. through Sample and Hold. ADC converts the analog data into digital form and puts it on the data bus. However, if the input is digital it can be given directly to the data bus, by-passing the S/H and ADC. Memory addressing unit gives the address of the memory location, where the input data is required to be stored.

Once the input data is available in the digital form, to be stored in the memory, an indication is required to be given to timing and control unit. This unit after getting the indication, puts the microprocessor in Hold for two states (T_{4H} and T_{5H}) of OF machine cycle. During these two states, appropriate control signals are also generated to enable the tristate buffers connected at the output of memory addressing and data conversion units. Thus, transparent DMA operation is performed during OF machine cycle.

An EPROM is provided to store the user's program. A DAC is connected at one of the output ports of SDK-85 kit to convert the processed digital data into analog form.

3.3 TIMING AND CONTROL UNIT:

This unit has two main functions:

a) To put microprocessor in HOLD during OF for two clock cycles - T_{4H} and T_{5H}.

b) During HOLD state generate various control signals.

Falling edge of sampling pulse has been used to indicate the EOC for the following reasons. During the high level of sampling period, S/H samples the input signal, while during low level it holds the signal. ADC has been connected in such a way, that the moment digital data is read from it, automatically the new conversion starts. Therefore from ADC, digital data should be read at the falling edge of the pulse, so that new conversion could start up during the Hold period of S/H circuit.

A JK flip-flop has been used as shown in Fig. 3.2 to keep the HOLD input of microprocessor high during T_1, T_2 and T_3 states of OF. It is falling edge triggered and has been used in toggle mode. In the beginning it is cleared by a pulse generated through the software. At the falling edge of the sampling pulse it gets set, this in turn raises Hold input high during OF machine cycle. At the end of T_3 state it gets reset, by HLDA and C_1+C_2 combination and thus HOLD is pulled low.

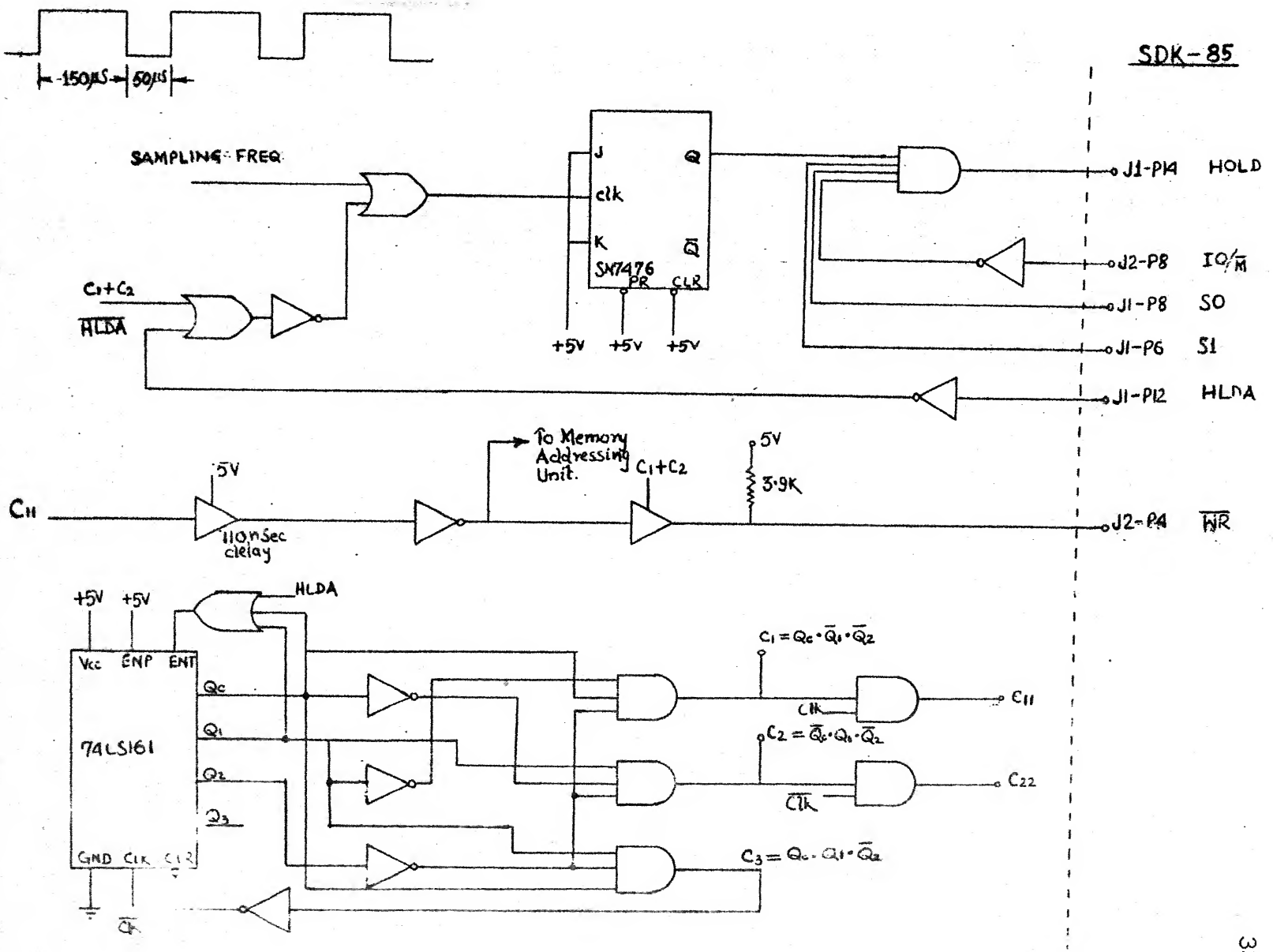


FIG 3·2 TIMING AND CONTROL UNIT

A four bit binary counter is used to generate various control signals during Hold period. HLDA enables the counter. Inverted system clock has been used as the clock for the counter. C_1 and C_2 outputs go high during T_{4H} and T_{5H} respectively.

3.4 MEMORY ADDRESSING UNIT:

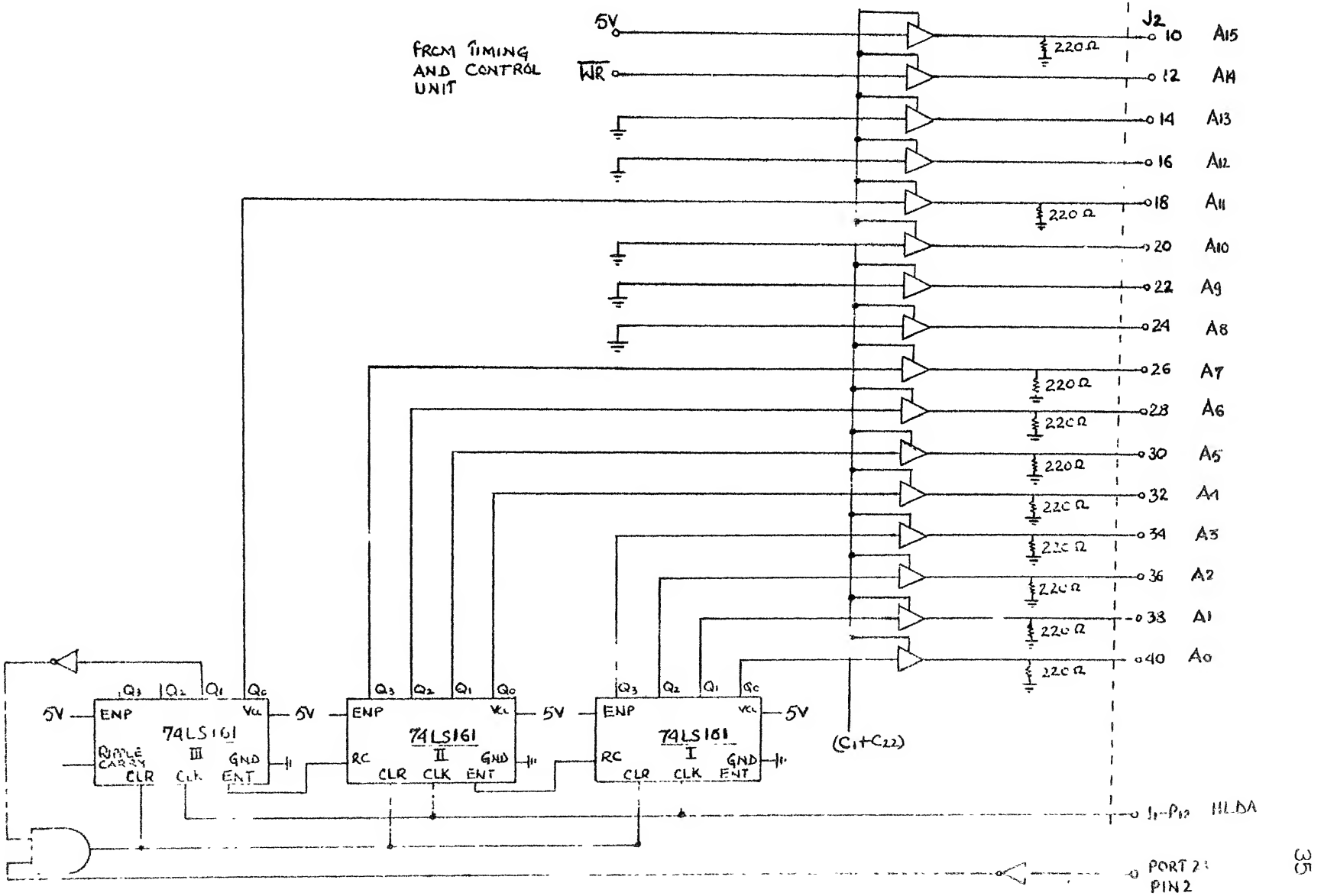
As the name implies, this unit generates the address of the memory location where input data is required to be stored. For two memory areas 8000-80FF and 8800-88FF the addresses are generated.

Three, four bit binary counters are connected in cascade form as shown in Fig. 3.3 to generate address bits. These are rising edge triggered and HLDA is given as the clock. During the T_3 state of OF when HLDA goes high, the count changes, and thus the address. However, the address is put on the bus only during T_{4H} and T_{5H} states and that is controlled by enabling the tristate buffers by C_1+C_{22}.

Bits A_8-A_{10}, A_{12}, A_{13}, and A_{15} are kept fixed, because they are common in both the memory areas. For A_{14}, $\overline{WR}$ signal is used, and reason for the same is given in Section 3.7.

To have automatic change over between the two memory areas Q_1 O/P of III counter is connected to clear input of the

FIG 3·3 MEMORY ADDRESSING UNIT



counters. Initially the counters are cleared software wise through port 23. Addressing range can be extended by making minor changes in output connections of III counter.

3.5 DATA CONVERSION UNIT:

Input analog signal is connected to Sample and Hold chip NE5537 as shown in Fig. 3.4. For proper logic operation, one of the logic pins must always be atleast 2V below the positive supply and 3V above the negative supply. Accordingly, high and low logic levels are fixed at 5V and OV respectively. High logic samples the signal, while low logic holds it.

AD7574 is a 8-bit microprocessor compatible ADC which uses the successive approximation technique to provide conversion time of 15 μsec. Table 3.1 and Fig. 3.5 show the truth table and timing requirements for interfacing the AD7574 like Read Only Memory. It has been used in ROM mode.

$\overline{CS}$ is held low permanently and converter operation is controlled by the $\overline{RD}$ input. The AD7574 $\overline{RD}$ input has been given as $\overline{HLDA}$. Thus data READ is initiated every time the microprocessor is put in HOLD. The converter is automatically restarted when $\overline{RD}$ returns high i.e. $\overline{HLDA}$.

$\overline{BUSY}$ High indicates that conversion is complete. $\overline{BUSY}$ must be High before a data READ is attempted, i.e. the total delay between a convert start and a data Read must be

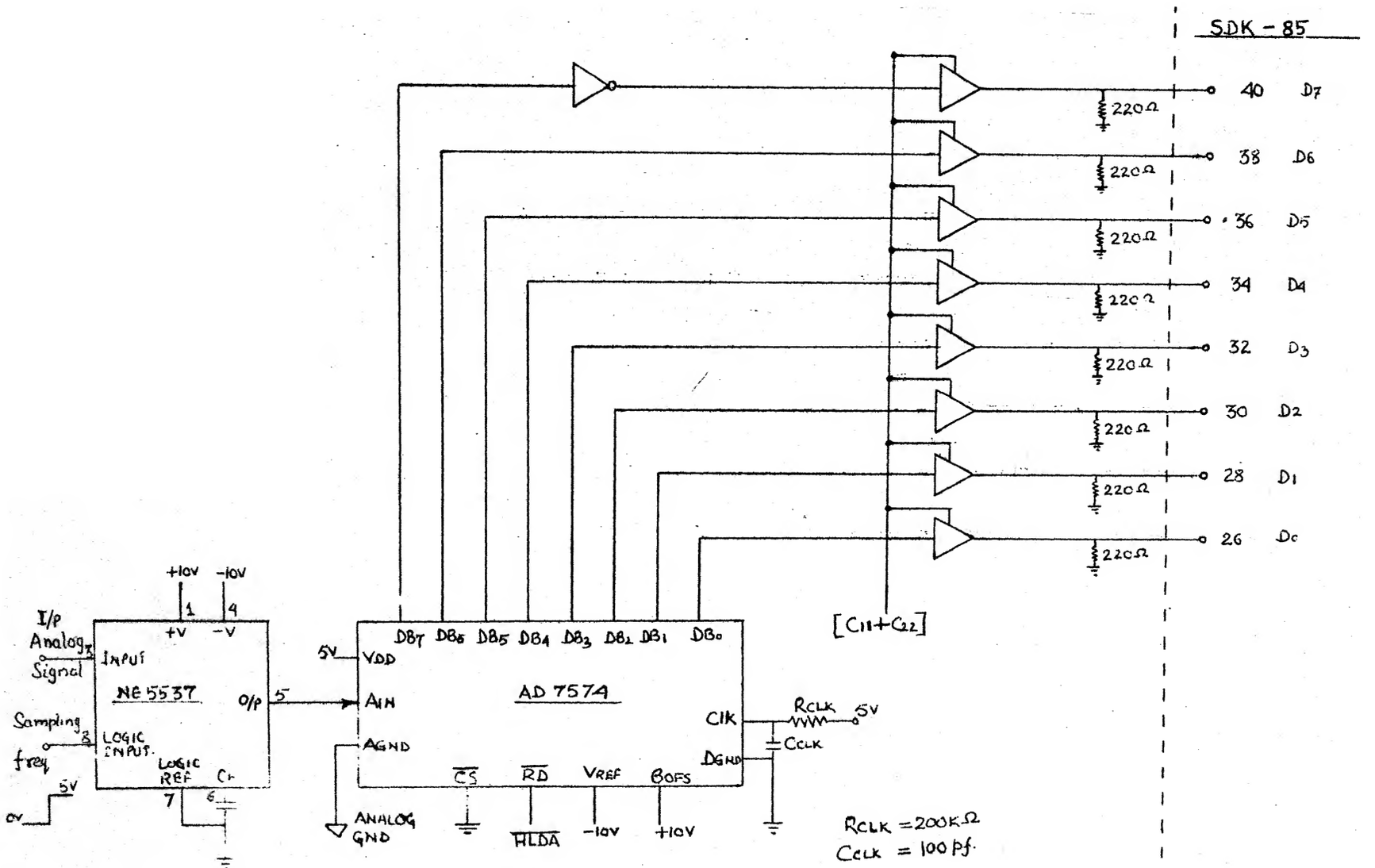
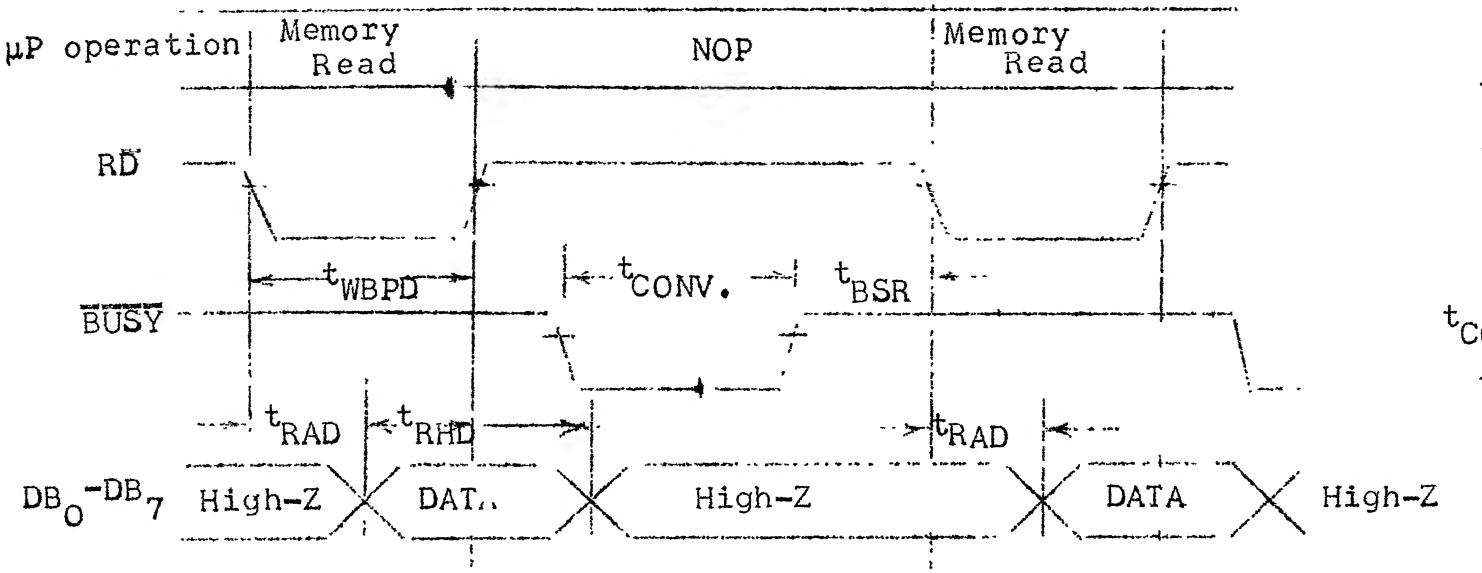


FIG 3·4 DATA CONVERSION UNIT



t_{RAD} Data Access time 220 nsec.

t_{RHD} Data Hold time 180 nsec.

t_{WBPD} $\overline{RD}$ High to $\overline{BUSY}$ Propagation dela 1.0 µsec.

$t_{CONV.}$ Conversion time 22 µsec.

t_{BSR} $\overline{BUSY}$ to $\overline{RD}$ low setup time

$\overline{RD}$ can go low pri to $\overline{BUSY}$=HIGH, but must not return H. until $\overline{BUSY}$ = HIGH

Fig. 3.5: ROM MODE TIMING DIAGRAM ($\overline{CS}$ HELD LOW)

AD7574 Inputs		AD7574 Outputs		AD7574 Operation
$\overline{CS}$	$\overline{RD}$	$\overline{BUSY}$	DB_0-DB_7	
L	⌐_⌐	H	High-Z to DATA	Data Read
L	_⌐_	⌐_	DATA to High-Z	Reset and Start new conversion
L	⌐_	L	High-Z	No effect, converter BUSY
L	_⌐	L	High-Z	Not allowed, causes incorrect conversion

Table 3.1: TRUTH TABLE ROM MODE

at least as great as the AD7574 conversion time. In the present set up as the data is read only after 200 μsec (sample period) which is much higher than the conversion time, $\overline{BUSY}$ has been ignored.

AD7574 has an internal asynchronous clock oscillator which starts upon receipt of a converter start command and ceases oscillating when conversion is complete. The clock requires an external R and C as shown in Fig. 3.4. Conversion time depends upon the R_{CLK} and C_{CLK} values. Here the conversion time is set as 22 μsec.

The AD7574 has been used in the Bipolar offset binary configuration. Fig. 3.6 illustrates the transfer characteristic for bipolar operation.

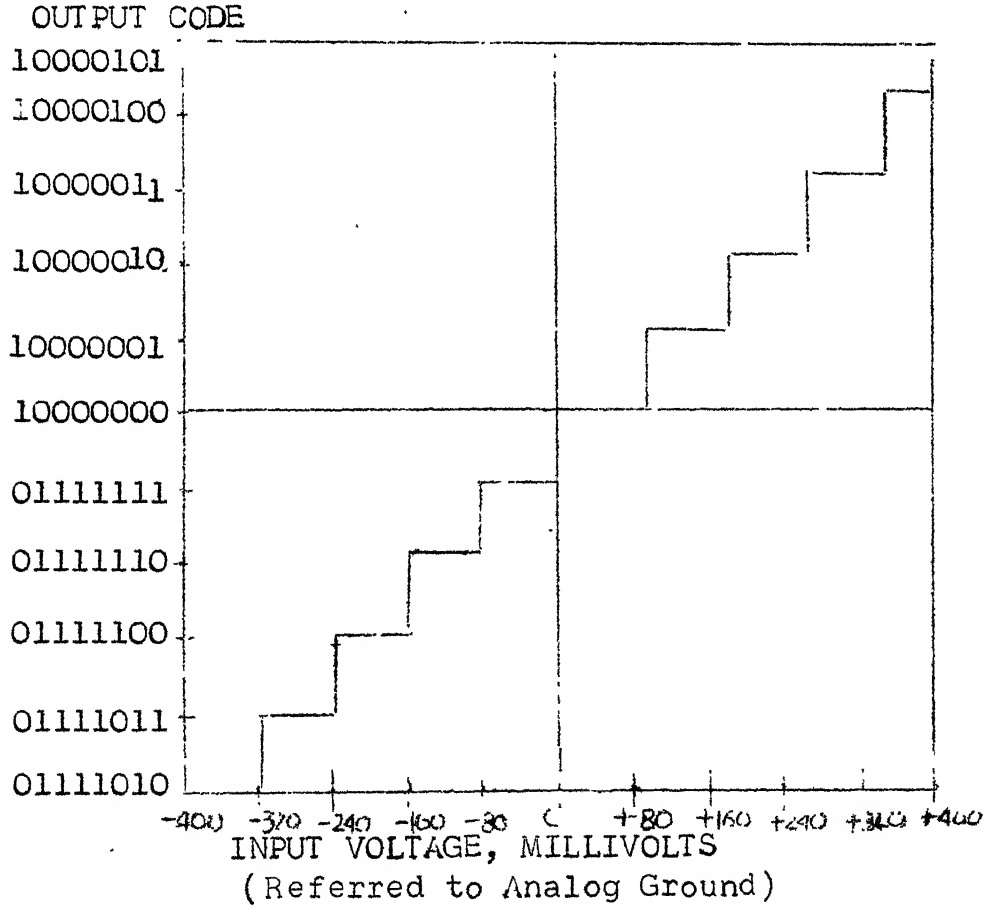


Fig. 3.6

In this configuration M.S.B. is '1' for positive input voltage, whereas '0' for negative values. To get data in 2's complement form, M.S.B. has been inverted externally.

Output of the AD7574 is connected to system data bus through tristate buffers. $C_{11} + C_{22}$ enable the buffer during T_{4H} and T_{5H} states.

3.6 DATA STORAGE UNIT:

This unit consists of two memory blocks to store input data, and one address decoder 74LS138 (Fig. 3.7). 2114AL-1 is a 1K-4 bit RAM. For 8 bit data two of these have been connected together in each block. These RAMs have been connected on the system buses and can be accessed by both microprocessor and the external hardware. 2114AL-1 timing details and pin configuration etc. have been given in Chapter 2.

Decoder 74LS138 chip (one out-of-8 decoder) decodes memory address bits to provide chip enables for the 2114AL-1 and an EPROM 2716. EPROM is used to store the Filter program. Table 3.2 lists each chip enable output,accompanied

SDK-85

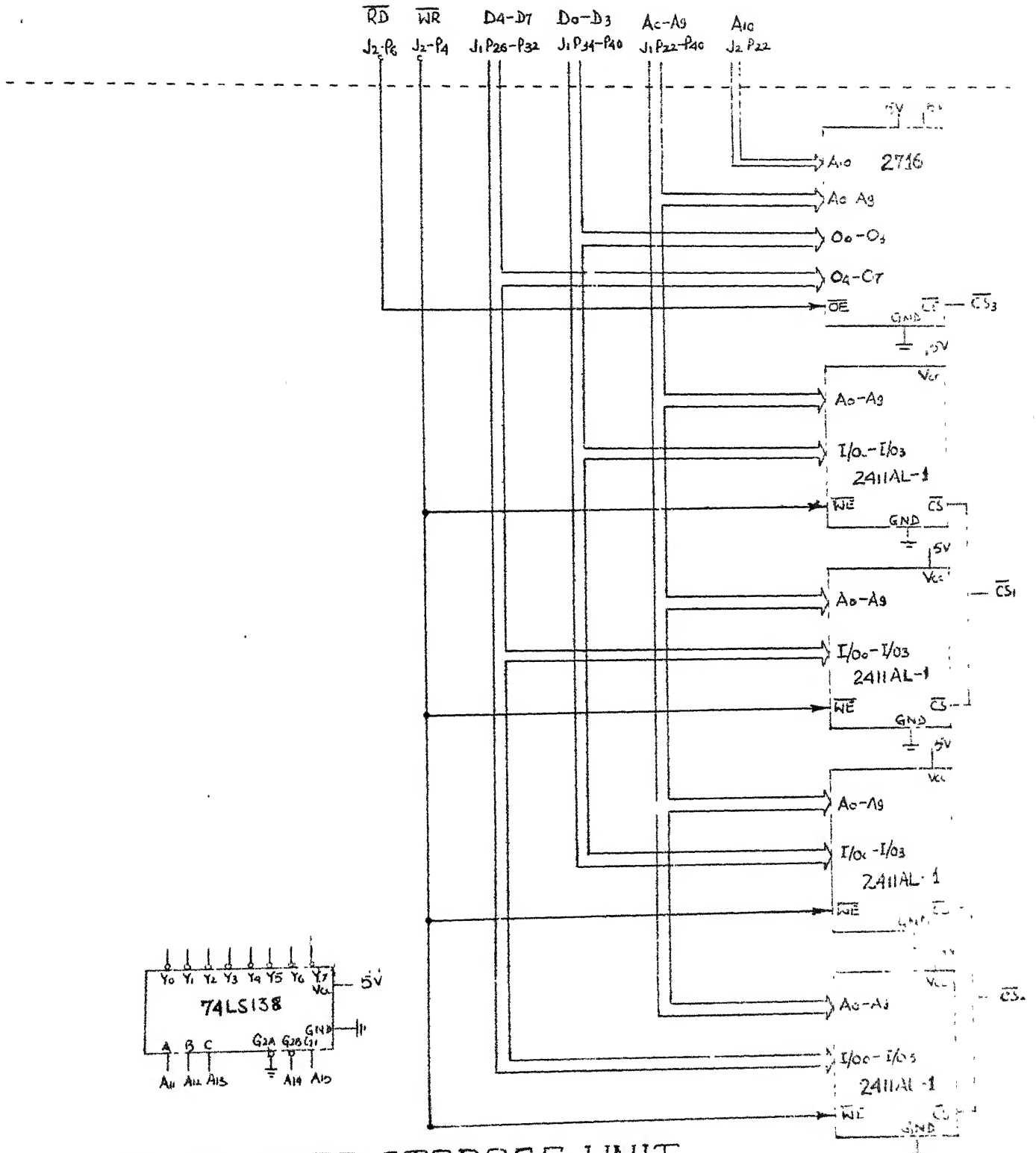


FIG 3·7 DATA STORAGE UNIT

by the address space over which it is active and the device that is selected.

Table 3.2: 74LS138 Chip Enables

OUTPUT	ACTIVE ADDRESS RANGE	SELECTED DEVICE
CS0	8000 - 87FF	2114AL-1 I BLOCK
CS1	8800 - 8FFF	2114AL-1 II BLOCK
CS2	9000 - 97FF	2716
CS3	9800 - 9FFF	N/C
CS4	A000 - A7FF	N/C
CS5	A800 - AFFF	N/C
CS6	B000 - B7FF	N/C
CS7	B800 - BFFF	N/C

N/C NOT Connected - Available for expansion

The above chip enable table can be expanded to form a memory map that illustrates the active portion of memory (Fig. 3.8 refers). The areas marked 'FOLD BACK' in Fig. 3.8 indicate address space that is unused, but available for expansion, because these locations are multiple mappings of the basic locations.

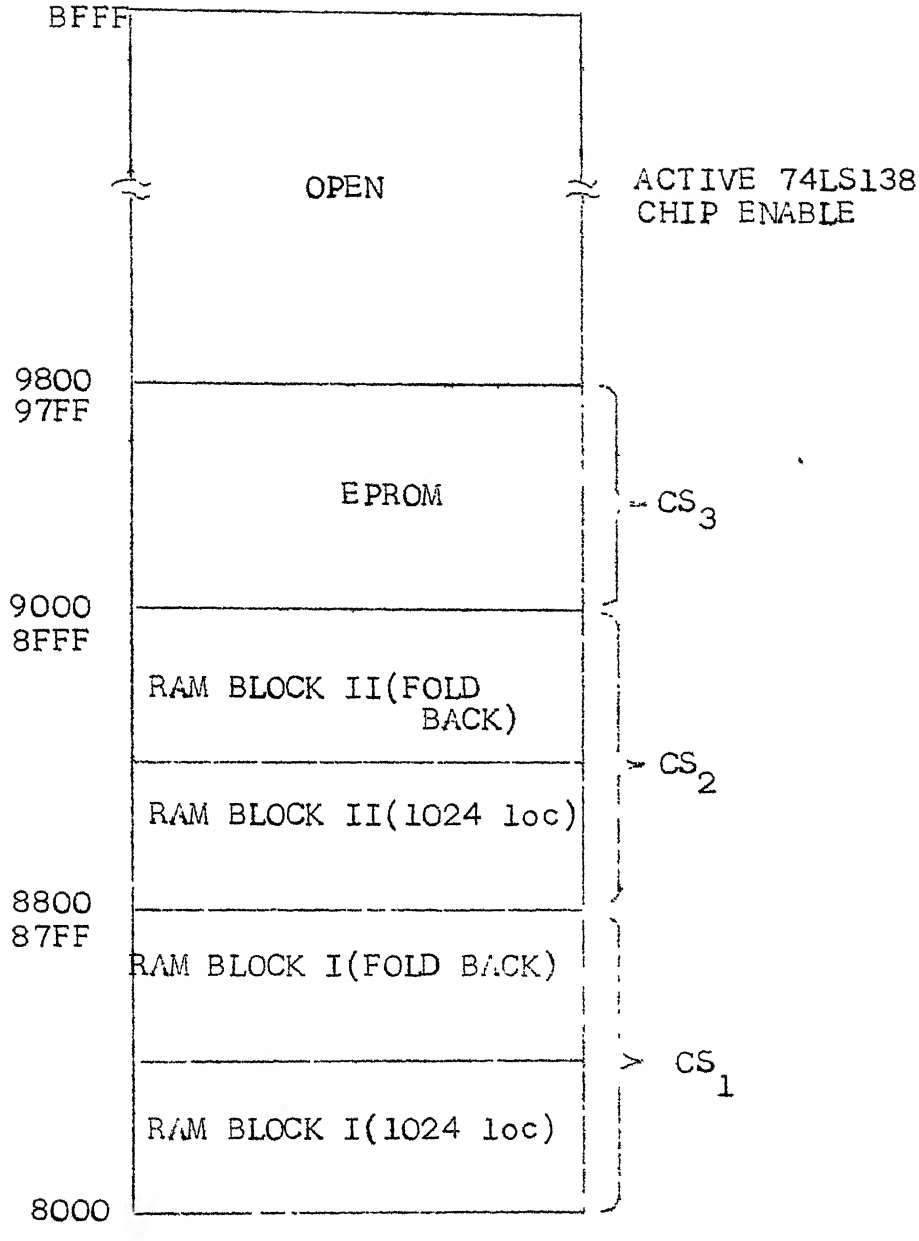


Fig. 3.8: MEMORY MAP

Areas marked 'OPEN' in Fig. 3.8 are free for expansion. Extra RAM and EPROM chips can be mounted. 74LS138 address decoder has 5 uncommitted chip select lines and can be used for memory expansion if required.

3.7 SOME SPECIAL HARDWARE DESIGN FEATURES:

i) Refer Fig.3.7. The address decoder circuit takes into account a peculiarity of the 2114AL-1 (RAM) chip. In the RAM 'write' cycle, if the $\overline{WE}$ ($\overline{WRITE\ ENABLE}$) remains high at the low going edge of the $\overline{CS}$, then the 2114AL-1 starts driving data through its output buffer. This of course, leads to bus conflict with the incoming data on the system bus. To avoid this, the $\overline{WE}$ has to simultaneously become low with $\overline{CS}$. Then the Dout lines become Hi-impedance and data can safely be written into RAM.

This has been done by giving externally generated $\overline{WR}$ to A_{14}. A_{14} is used as one of the enable inputs for 74LS138. This way $\overline{WE}$ and $\overline{CS}$ go low simultaneously.

ii) Refer Fig. 3.3 and 3.4. Memory addressing and data conversion units are connected to the system buses directly, and give TTL outputs. Fall time of the address and data bits are required to be minimised, so that they can be accomodated within T4H and T5H states. For the optimum reduction in the fall time, 220 ohm resistors are used for each line.

iii) To introduce delay in the control signals 74126 buffers have been used. Each buffer gives 10 nsec delay. Delays are necessiated to meet the timing requirements.

iv) As a precautionary measure, wherever in the circuit MOS output is required to drive more than one TTL inputs, it is buffered. For this purpose 74126 tristate buffers as well as ordinary TTL inverters have been used, depending on the particular application.

v) While selecting the ICs prime consideration has been the speed and the cost. Table 3.3 gives a list of the ICs used.

Table 3.3: List of ICs Used

Device Type	Delay (in nsec)	Description
SN74126	10.0	Quad bus buffer gate with Tri-state output
SN74S04	3.0	Hex Inverter
SN74S11	4.75	Triple three input positive AND gate
SN74S32	4.0	Quad two input positive OR gate
SN74LS161	14.0	Synchronous 4-bit binary counter
SN74LS138	22.0	Address decoder
SN7476		Dual JK flip-flop
2114AL-1		1024x4 bit RAM
AD7574		8-bit ADC
NE 5537		Sample and Hold
DAC 08BC		8-bit DAC
2716		EPROM (2Kx8 bits)

CHAPTER 4

DESIGN OF SOFTWARE FOR DIGITAL FILTER

4.1 INTRODUCTION:

The field of digital signal processing has grown enormously in the past two decades. The major sub-divisions of the field of digital signal processing are digital filtering and spectrum analysis. The field of digital filter is divided into Finite Impulse Response (FIR) filters and Infinite Impuse Response (IIR) filters.

A low pass sencend order Butterworth digital filter has been implemented (software wise) on the Transparent DMA data acquisition system to demonstrate its application in the field of digital signal processing.

4.2 DIGITAL FILTER DESIGN FROM CONTINUOUS - TIME FILTERS:

The filter design problem is basically a mathematical approximation problem. The domain in which the approximation problem is solved determines how and where the resulting filter can be used. Thus if the approximation problem is solved in the Z-plane, the resulting filter is a digital filter. If it is solved in the S-plane, the resulting filter is an analog filter.

Simple mapping procedures can be used to transform filters from one domain to filters in the other domain. This technique of designing an appropriate continuous-time filter and digitizing the resulting design to give a digital filter is the most popular IIR design technique and is most useful for designing standard filters such as Lowpass, Bandpass, Bandstop, and High pass filters where a considerable body of theory on continuous-time filters is available. There are four most widely used procedures for digitizing the transfer function of an analog filter. They are -

i) The method of mapping of differentials
ii) The impulse invariant transformation
iii) The bilinear transformation
iv) The matched Z-transform technique

Having discussed the basic principle of the design of digital filters, first the transfer function of a low pass second order Butterworth filter with cutoff frequency 1.12 KHz is determined. Then by using the Impulse Invariant Transformation technique analog filter is digitized by sampling period of 200 μsec (sampling frequency 5 KHz) and transformed from S-plane to Z-plane. Digital filter's expression in the Z-plane is then transformed in the time domain to get

the final expression for which the program has been written and implemented on the designed system.

4.3 ANALOG LOW PASS BUTTERWORTH FILTER:

Butterworth lowpass filters are characterized by the property that the magnitude characteristic is maximally flat at the origin of the S-plane. The squared magnitude function $|T(j\omega)|^2$ of a Butterworth filter is

$$|T(j\omega)|^2 = \frac{1}{1+c^2(\frac{\omega}{\omega p})^{2n}} \qquad (4.1)$$

where ω should be interpreted as the frequency normalised with respect to the pass band edge ωp.

c - error criterion

n - is the filter order

The frequency response of a Butterworth filter for various values of n is shown in Fig. 4.1. In order to realize the filler we have to determine its transfer function. The poles of an all all-pole transfer function (i.e. all the transmission zeros are at infinity) must be in the left half plane to satisfy the realizability condition. If we let

$$|T(j\omega)|^2 = \frac{1}{|H(j\omega)|^2}$$

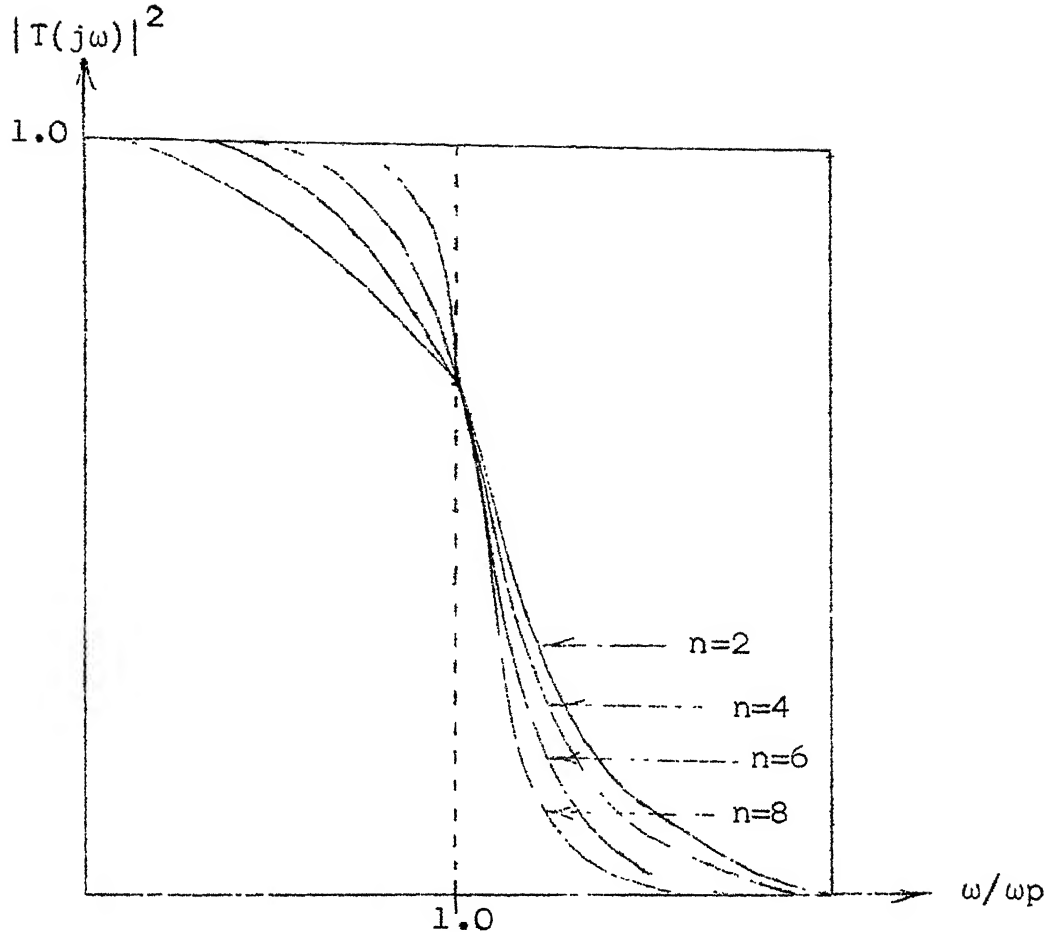


Fig. 4.1: Frequency Response of Butterworth Filter

and we can write from equation (4.1) for n=2 (second order)

$$|H(j\omega)|^2 = H(s)H(-s)\big|_{s=j\omega}$$

$$= 1 + c^2 \left(\frac{\omega}{\omega p}\right)^4$$

$$= 1 + c^2 \frac{(-s^2)^2}{(\omega p)^4}$$

The 4 zeros of H(s) H(-s) are given by solving

$$1 + c^2 \frac{s^4}{\omega p^4} = 0$$

$$\text{or} \quad S^4 = - \frac{\omega p^4}{c^2}$$

The 4 roots are

$$\frac{\omega p}{\sqrt{c}} e^{jk\pi/4} \qquad k = 1,3,5,7$$

Values of k=3 and 5 give the LHP (left half plane) roots

$$(s - \frac{\omega p}{\sqrt{c}} e^{j135^\circ}) (s - \frac{\omega p}{\sqrt{c}} e^{j225^\circ})$$

$$= (s^2 + \frac{2\omega p}{\sqrt{c}} \cdot \frac{1}{\sqrt{2}} \cdot s + \frac{\omega p^2}{c})$$

$$H(s) = (s^2 + \sqrt{2} \frac{\omega p}{\sqrt{c}} \cdot s + \frac{\omega p^2}{c})$$

$$T(s) = \frac{1}{(s^2 + \sqrt{2} \frac{\omega p}{\sqrt{c}} \cdot s + \frac{\omega p^2}{c})}$$

Let c=1 and ωp at 3 db point = 7070 Rad/sec = $2\pi fp$

fp = 1125 Hz.

$$\text{So,} \quad T(s) = \frac{1}{s^2+\sqrt{2}.7070.s+(7070)^2}$$

$$= \frac{1}{s^2+10,000s+(\frac{10,000}{\sqrt{2}})^2}$$

$$= \frac{1}{(s-5000(-1+j))(s-5000(-1-j)}$$

Taking partial fraction and solving the above expression

$$T(s) = \frac{-j}{10,000}\left[\frac{1}{s-5000(-1+j)} - \frac{1}{s-5000(-1-j)}\right]$$

Thus, the transfer function of the analog low pass second order Butterworth filter having cut off frequency 1.125 KHz is

$$T(s) = \frac{-j}{10,000}\left[\frac{1}{s+5000(1-j)} - \frac{1}{s+5000(1+j)}\right] \qquad (4.2)$$

4.4 IMPULSE INVARIANT TRANSFORMATION TECHNIQUE:

The characteristic property of this transformation is that the impulse response of the resulting digital filter is a sampled version of the impulse response of the analog filter. In consequence of this result, the frequency response of the digital filter is an aliased version of the frequency response of the corresponding analog filter.

Mapping relation of this technique for transfer function H(s) from S-plane to Z-plane, Z-transform is

$$\frac{1}{s+di} \longrightarrow \frac{1}{1-z^{-1}e^{diT}} \qquad (4.3)$$

where T - is the sampling period

di - is the Filter Coefficient

By direct application of the above mapping relation (eqn.(4.3)) to the analog filter's transfer function T(s) (eqn.(4.2) we get Z-transform

$$H(z) = \frac{-j}{10^4}\left[\frac{1}{1-z^{-1}.e^{-5000(1-j)T}} - \frac{1}{1-z^{-1}\ e^{-5000(1+j)T}}\right]$$

$$= \frac{-j}{10^4}\left[\frac{z^{-1}\ e^{-5000T}(-e^{-j5000T} + e^{j5000T})}{(1-z^{-1}.e^{-5000(1-j)T})(1-z^{-1}e^{-5000(1+j)T}}\right]$$

$$= \frac{-j}{10^4}\left[\frac{e^{-5000T}.z^{-1}.2j\ \sin 5000T}{1-e^{-5000T}.z^{-1}.2\cos 5000T+e^{-10000T}z^{-2}}\right] \qquad (4.4)$$

$$H(z) = \frac{Y(z)}{X(z)} \qquad \text{where } Y(z) \text{ - output}$$

$$X(z) \text{ - Input}$$

thus equation (4.4) becomes

$$Y(z)[1-e^{-5000T}.\ z^{-1}.2.\cos 5000T+e^{-10000T}.z^{-2}]$$

$$= X(z)\ [\frac{2}{10^4}\ .\ e^{-5000T}.\ z^{-1}\ .\ \sin 5000T] \qquad (4.5)$$

Converting this expression from Z-transform to time domain,

$$Y(n) - Y(n-1).e^{-5000T}.2\cos 5000T+Y(n-2)e^{-10000T}$$

$$= X(n-1)\ .\ \frac{2}{10^4}\ .e^{-5000T}.\ \sin 5000T$$

Taking sampling frequency as 5 KHz

sampling period T = 200 μsec.

$$5000T = 5000x200x10^{-6} = 1$$

$$Y(n)-Y(n-1).e^{-1}.2\cos 1 + Y(n-2)e^{-2}= \frac{2}{10^4}\ .\ e^{-1}.\sin 1.\ X(n-1)$$

$$= X'(n-1)$$

Thus, the expression for the low pass second order Butterworth digital filter (with cutoff frequency 1.125 KHz and sampling period 200 μsec) is

$$Y(n) = X'(n-1)+Y(n-1)0.4 - Y(n-2)\ 0.14 \qquad (4.6)$$

4.5 DIGITAL FILTER SOFTWARE:

A program has been written to compute the value Y(n) for the following expression of low pass Butterworth filter.

$$Y(n) = X'(n-1) + Y(n-1)0.4 - Y(n-2)0.14$$

The program is executed repeatedly and continuously, thereby imitating the behaviour of the filter on a sampled basis. Program listing has been placed from page 57 to 60.

As has been mentioned in Chapter 3 that, there are two memory blocks where the input data is stored. We call them as block A and B(Fig. 4.2).

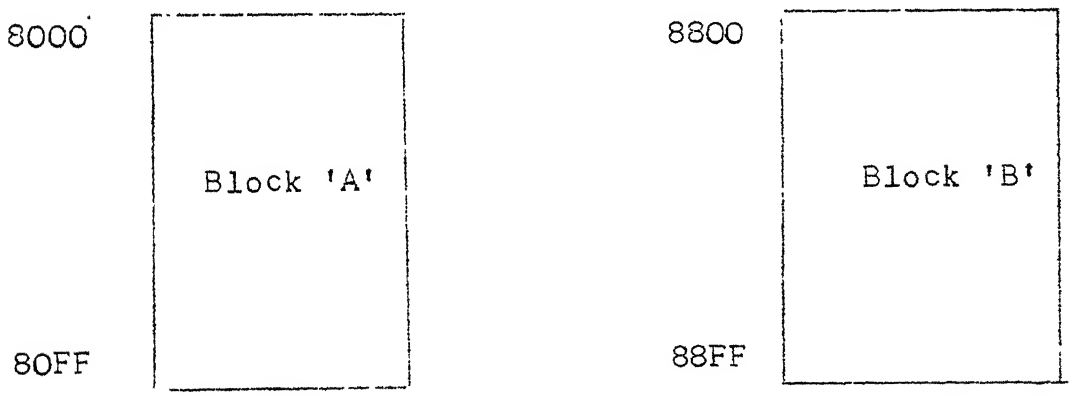


Fig. 4.2:

Each block has capacity of 256 bytes. Initially when the program execution starts, it waits in the 'WAIT' loop till the time block 'A' gets filled up completely. At this

stage program starts processing the Input data from block 'A'. While this process is going on input data starts getting into the next block i.e. B. By the time processing of block A finishes, B block gets filled up completely, with the input data. Then processing starts up for B-block and input data starts getting into A-block. This process of change over is automatic and continuous.

There are two 8155 RAM memory chips on SDK-85 kit, each RAM has three IN/OUT ports. These ports have been used to output the data, read the current value of the externally generated address etc.

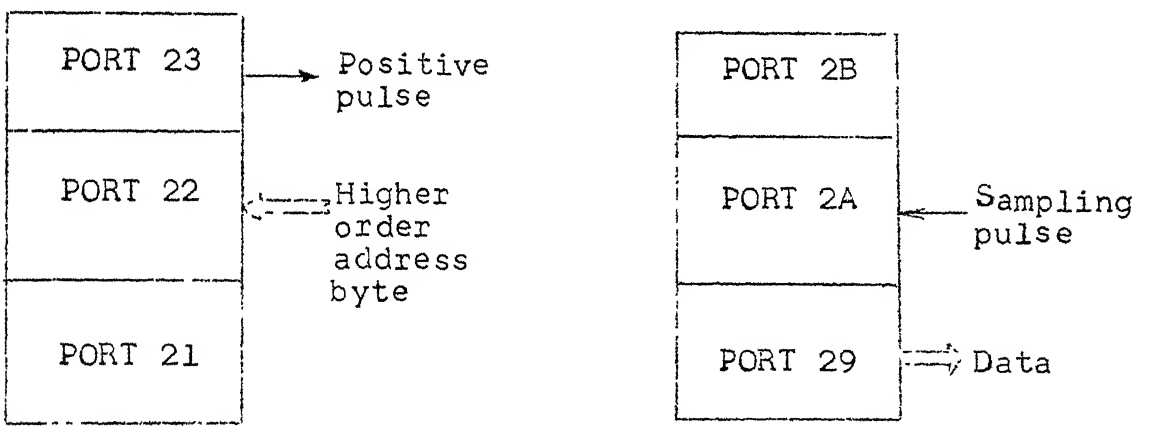


Fig. 4.3: IN/OUT Ports of SDK-85 Kit

For clear input of various counters and flip-flops used in the external hardware a positive pulse is generated through software and after inversion it is used. For this Port 23 has been used. To output the value of Y(n) Port 29 has been used.

To synchronise the output with the input, rising edge of the sampling frequency has been used in the software as reference. This way, output data i.e. Y(n) is available at the output port exactly after every 200 μsec. For this purpose sampling pulse has been taken in from port 2A.

SYMBOLIC		COMMENTS
LXI	SP, 20C2	; Initialize Stack Pointer
MVI	A, OC	; Put 8155 Command in A Register
OUT	20	; Program the 8155 CSR
MVI	A, 01	; Put 8155(Extn) Command in A Register
OUT	28	; Program the 8155 (Extn) CSR
MVI	A, 00	; Output one positive pulse
OUT	23	; through Port 23 to clear
MVI	A,FF	; the counters and flip-flop
OUT	23	;
MVI	A, 00	;
OUT	23	;
STA	2002	; Store initial value of Y(n-1) as 00 at loc. 2002.
STA	2003	; Store initial,value of Y(n-2) as 00 at loc. 2003
STA	8000	; Store initial value of X(n-1) as 00 at loc 8000
STA	2006	; Store address of X(n-1) at loc
MVI	A,80	; 2006 and 2007, initial value
STA	2007	; as 8000.
WAIT: IN	22	; Wait in the loop till the time
XRI	88	; first block of memory i.e. 8000-80FF
JNZ	WAIT	; is getting filled up.

```
CALCU: MVI  L,00    ; This segment of the program
       LDA  2002    ; loads the value of Y(n-1) in
       MOV  H,A     ; H-Register from Location 2002
       ARHL         ; and computes the value of
       ARHL         ; 0.4 Y(n-1) in 16 bits. The
       MOV  B,H     ; computed value of 0.4 Y(n-1)
       MOV  C,L     ; in 16-bits form is then
       ARHL         ; stored at location  2004
       MOV  D,H     ; and 2005.
       MOV  E,L     ;
       DAD  B       ;
       XCHG         ;
       ARHL         ;
       ARHL         ;
       ARHL         ;
       MOV  B,H     ;
       MOV  C,L     ;
       ARHL         ;
       DAD  B       ;
       DAD  D       ;
       SHLD 2004    ;
       MVI  L,00    ; This segment loads the value
       LDA  2003    ; of Y(n-2) in H-register from location
       MOV  H,A     ; 2003 and  computes the value of
       ARHL         ; 0.14Y(n-2) in 16-bits. The Computed
       ARHL         ; value of 0.14 Y(n-2) in 16 bits
       ARHL         ; form is then stored in B and C
       MOV  D,H     ; Registers.
       MOV  E,L     ;
       ARHL         ;
       ARHL         ;
```

```
        ARHL            ;
        ARHL            ;
        MOV   B,H       ;
        MOV   C,L       ;
        ARHL            ;
        DAD   B         ;
        DAD   D         ;
        MOV   B,H       ;
        MOV   C,L       ;
        LHLD  2004      ; Load 0.4 Y(n-1) in H and L
        DSUB            ; 0.4 Y(n-1)-0.14Y(n-2) Result in H and L
        MOV   B,H       ; Store Result of above substration in B Reg.
        LHLD  2006      ; Get address of X(n-1) in H and L.
        MOV   A,M       ; X(n-1) value in Accumulator
        ADD   B         ; 0.4Y(n-1)-0.14Y(n-2) added with X(n-1)
        MOV   B,A       ; computed value of Y(n) from A to B
LOOP:   IN    2A        ; wait in the loop till the time
        XRI   80        ; rising edge of the sampling
        JNZ   LOOP      ; pulse comes.
        MOV   A,B       ; Bring Y(n) in Accr.
        OUT   29        ; output Y(n) to Port 29.
        STA   2001      ; Store Y(n) at loc. 2001
        MOV   A,L       ; Check whether all the 256
        XRI   FF        ; input samples X(n-1) from one block
        JNZ   CONT      ; have been processed or not.
        MOV   A,H       ; This segment changes over
        XRI   88        ; from one memory block to
        JZ    SKIP      ; another.
        MVI   H,88      ;
        JMP   COMON     ;
```

```
SKIP : MVI  H, 80   ;
COMON: MVI  L, 00   ;
       SHLD 2006    ;
       JMP  NORM    ;
CONT : INR  L       ; Increment X(n-1)'s address count
       SHLD 2006    ; by one.
NORM : LDA  2002    ; Shift Y(n-1) to Y(n-2) loc.
       STA  2003    ;
       LDA  2001    ; Shift Y(n) to Y(n-1) loc.
       STA  2002    ;
       JMP  CALCU   ; Jump for next X(n-1) computation
```

The execution time of the program is 171 μsec. However the computed value of Y(n) is available at the output through Port 29 only after 200 μsec as it has been synchronised with the sampling pulse.

4.6 OBSERVATION:

At the input of the Transparent DMA data acquisition system a sine wave signal has been given of fixed amplitude 5 volts. Sampling frequency has been kept constant at 5 KHz, and the input signal frequency is varied from 0 Hz to 5 KHz. At the output of the digital filter signal level and the frequency has been observed. For different input signal frequencies the variation in the output amplitude and frequency have been plotted (Fig. 4.4 and 4.5 refers).

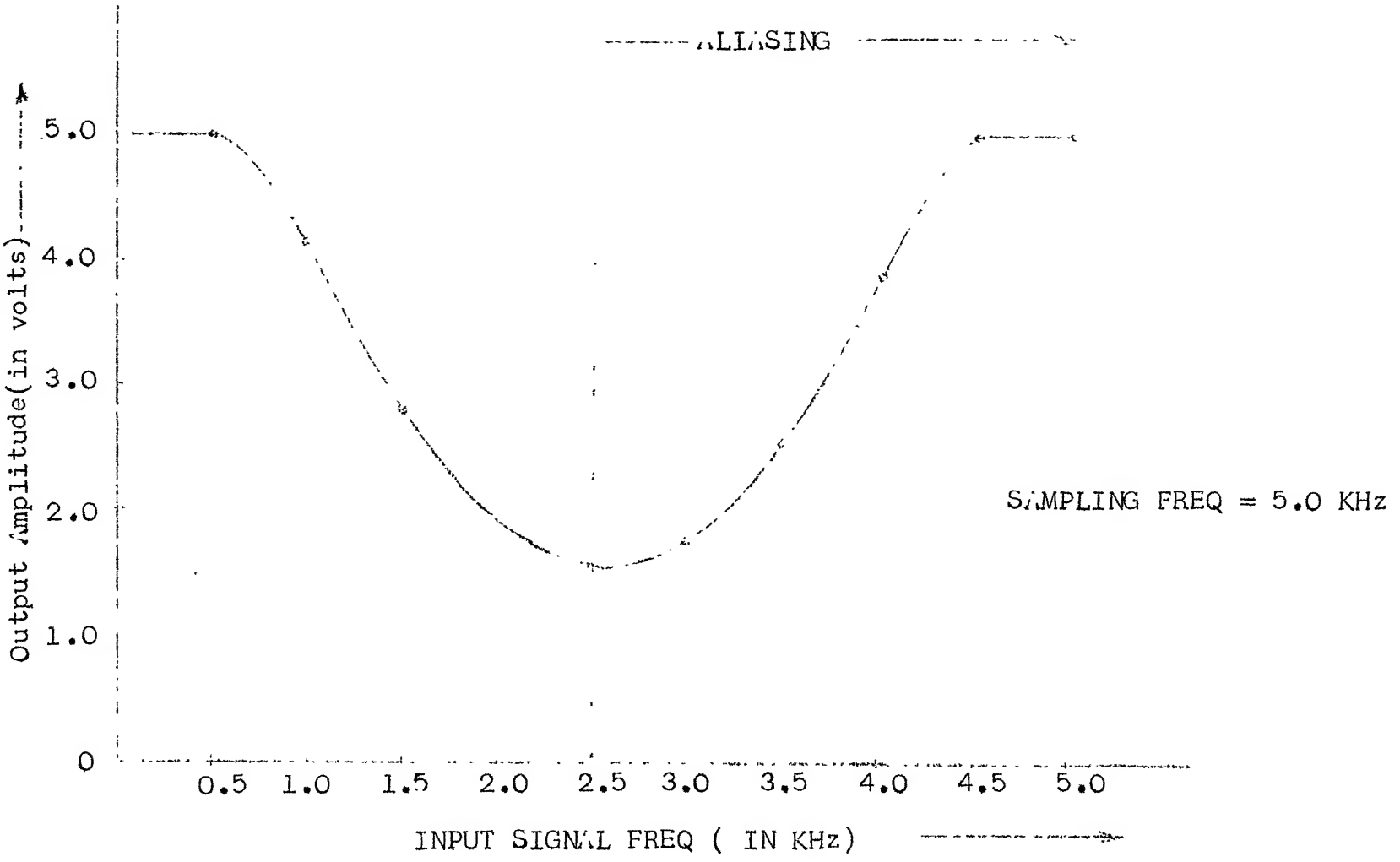

ALIASING
Output Amplitude(in volts)
5.0
4.0
3.0
2.0
1.0
0
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
INPUT SIGNAL FREQ (IN KHz)
SAMPLING FREQ = 5.0 KHz

Fig. 4.4

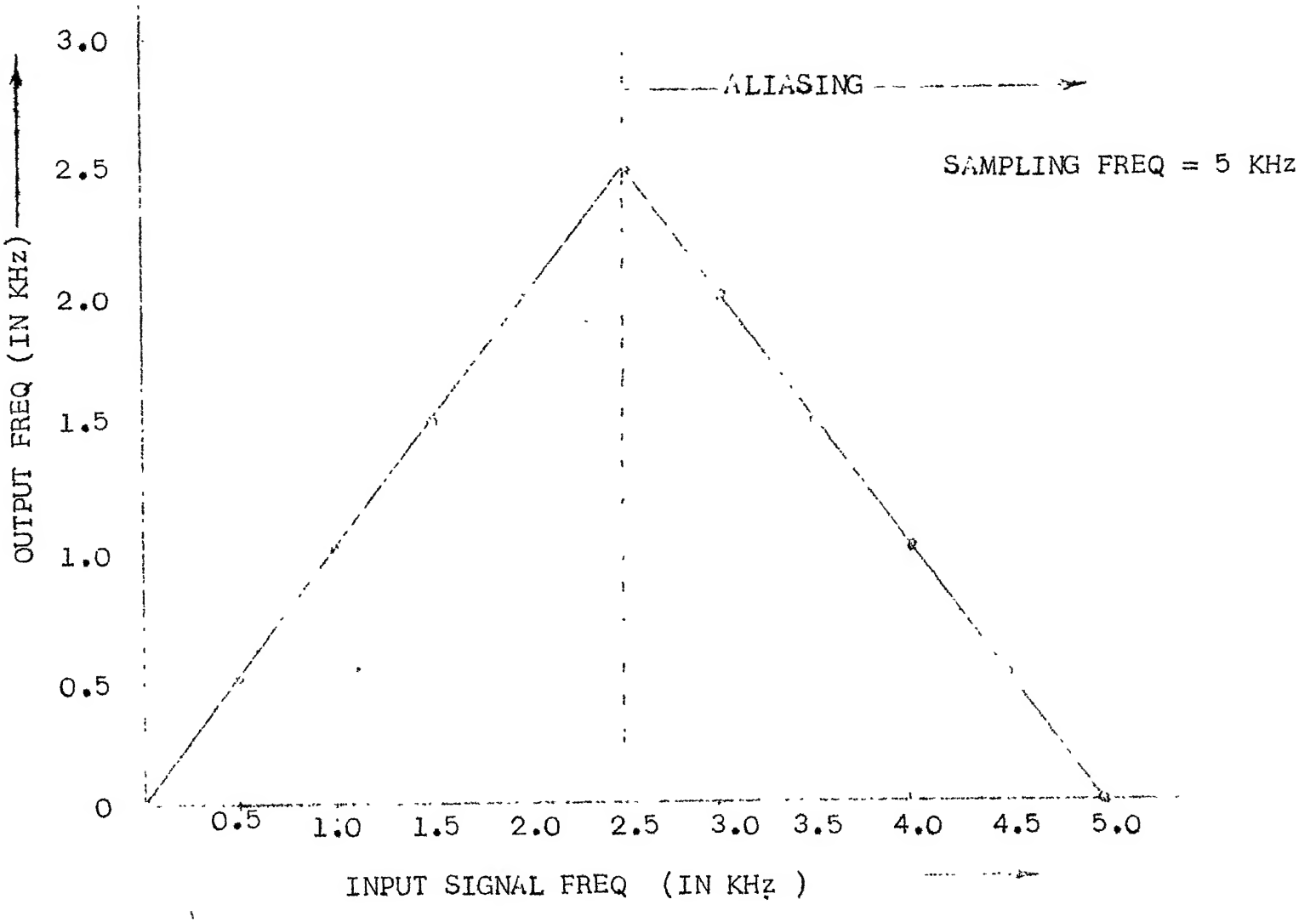

ALIASING
SAMPLING FREQ = 5 KHz
OUTPUT FREQ (IN KHz)
3.0
2.5
2.0
1.5
1.0
0.5
0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
INPUT SIGNAL FREQ (IN KHz)

Fig. 4.5

From the graph of Fig. 4.4 it can be seen that the output amplitude remains almost constant at 5.0V level upto 700 Hz. Then the output level drop is gradual upto the cutoff frequency of 1.125 KHz. However from 1.125 KHz to 2.5 KHz slope of the curve is rather steep. At 2.5 KHz (i.e. half of sampling frequency) the output level is least. Whereas frequency of the output signal remains the same as that of Input upto 2.5 KHz.

At the input of the digital filters de-aliasing filter is required to be put to avoid aliasing. But in this set up de-aliasing filter has not been put at the input to test aliasing. Aliasing has been observed at the output beyond 2.5 KHz.

As the input signal frequency is increased beyond 2.5 KHz output amplitude starts increasing and the frequency decreases. From 4.6 KHz to 5.0 KHz output level almost becomes constant at 5.0 volts. Then at 5 KHz it is observed the output is almost D.C.

CHAPTER 5

CONCLUSION

5.1 SUGGESTION FOR MODIFICATION AND IMPROVEMENTS:

(a) The designed transparent D.M.A. data acquisition system takes two clock states for one byte of data transfer. The data transfer can be done in only one state, i.e. T_4. A scheme to do the same is discussed briefly in the following paragraphs.

In this case, the microprocessor is required to be put in the Hold state only during T_4 state of OF. The Hold input of the microprocessor can be raised high like it has been done in the designed system, but should be pulled low, at the end of T_2 state. An additional counter is required for this purpose. The Hold line can be connected to the enable input of the counter, and inverted system clock to the clock input.

In the designed system HLDA and C_1+C_2 combination resets the JK flip-flop. Similarly here the counter's output logic can be used for resetting the flip-flop at the end of T_2 state thus, pulling the Hold input also low.

Access time of 2114A^L-1 RAM is 100 nsec. Timing for various control signals can be adjusted as shown in

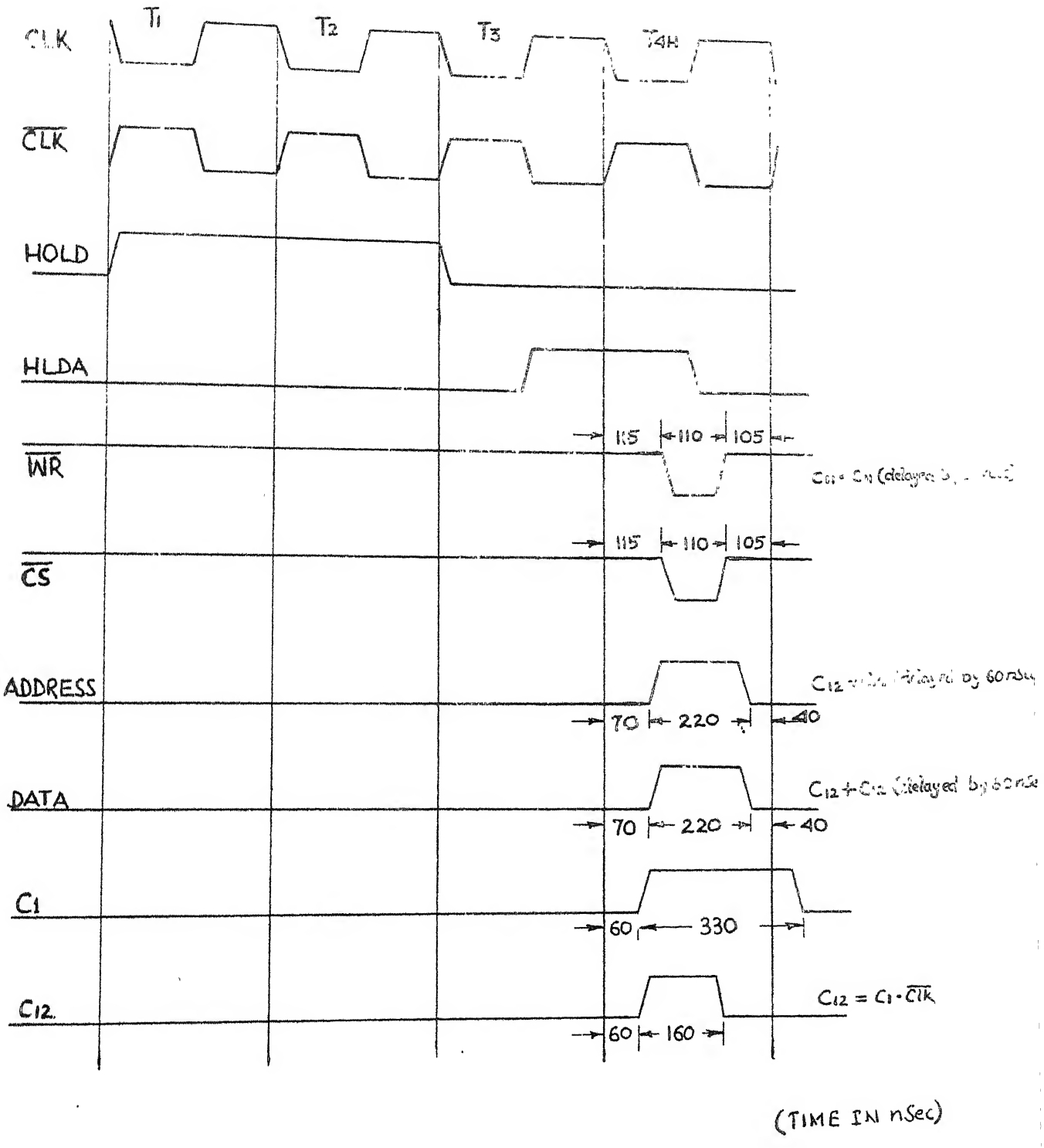


FIG 5·1 TIMING DIAGRAM : PROPOSD SYSTEM

by simulation, a collection of major analog modules, and their interconnections. A program is executed repeatedly and continuously, thereby imitating the behaviour of the analog system on a sampled basis.

Examples of such functions include simple to complex filters, oscillators, limitors, rectifiers, modulators, non-linear function, correlations, and logical operations among many others. Several such functions can be achieved by using a single 2920 chip, for extermely complex designs additional 2920S can be cascaded together.

It uses 5 MHz or 10 MHz clock, and performs 25 bit arithmetic. It has limited data storage, size of the RAM provided is only 40 words by 25 bits. User program also has the limit of 192 instructions.

5.3 Comparision between Designed System and 2920:

The designed system can be compared in certain respects with the 2920. It can perform all the functions which are performed by the 2920 as listed in the above section.

i) In the designed system data storage space 2Kx8 is much larger compared to 40x25 bits in the 2920. This storage space can further be increased by adding additional RAM chips, whereas in 2920 it is fixed.

ii) Similarly the 2920 has limited space in PROM for user's program. In the designed system 2K EPROM is provided for this purpose and additional EPROMS can also be used if required.

iii) In 2920 it is cumbersome to give input and get output in digital form, whereas the designed system accepts both analog and digital inputs and gives also in either form.

iv) In the 2920 system, maximum sampling rate that can be achieved with full 192 instructions program length is 6.5 KHz by using 5 MHz clock. The designed system with 3.125 MHz clock for 75 instructions program allows 5.75 KHz maximum sampling rate.

v) The 2920 is a very powerful chip but very expensive and requires special PROM burner. Therefore in certain specific areas where complex systems like 2920 are not needed, the designed system because of its flexibility can be very useful.

5.4 CONCLUSION:

The transparent DMA data acquisition system for real time applications is designed with the aim that during data acquisition, processor should not be interrupted. One of the signal processing functions has been tried out on the

designed system successfully. A program for low pass Butterworth filter is executed repeatedly and continuously, thereby imitating the behaviour of the analog filter on a sampled basis. During the execution of the program input data has been stored without interrupting the program execution.

The pass through a program establishes a sample interval i.e. input/output operation and takes place once per program pass. Low pass filter program execution takes 171 μsec thus 5.75 KHz maximum sampling rate is achieved. The sampling rate can further be increased, if the same system is tried out with more powerful microprocessor e.g. the 8086-1. The 8086-1 has built-in multiplication and division instructions and performs 16-bit arithmetic operations. It uses 10 MHz clock. The 8085-A microprocessor which has been used in this project, does not have multiplication and division instructions, and performs 8-bit arithmetic. To perform 16-bit arithmetic operations on 8085-A additional instructions are required and it's clock rate is only 3.1 MHz. Thus, we see that with better microprocessor, much higher sampling rate can be achieved for this system.

Transparent D.M.A. data acquisition system would be very useful in the field of real time applications - like digital signal processing, computer networks, automatic message switching systems and communication system monitoring etc.

REFERENCES

1. An Introduction to microcomputers, vol. I - Osborne Series.

2. An Introduction to microcomputers, vol. II - Osborne Series.

3. The Bipolar Digital Integrated Circuits Data Book, Texas Instruments.

4. Data Acquisition Component Handbook - DATEL INTERSIL (DAC).

5. Data Acquisition Data Book - Analog Devices (ADC).

6. Component Data Catalog - INTEL

7. Network Theory and Filter Design - Vasudev K. Aatre. Wiley Eastern Ltd. Meerut 08.

8. Theory and Application of Digital Signal Processing - L.R. Rabiner and Bernard Gold. Prentice-Hall India 1978.

9. SDK-85 System design kit User's Manual. 1979.